****

Addressing a Commercial Grade Operating System Security Functional Requirement Set with Windows Vista and Server 2008

Microsoft Corporation

Version 1.0 Published: December 2008

Version 2.0 Published: March 2009

Author: Mike Lai ([mikelai@microsoft.com](mailto:mikelai@microsoft.com))

Abstract

An operating system (OS) is critical for enabling general information technology (IT) services over a network for users. Because of the fundamental resources being consumed by the IT services, the OS must enforce appropriate security policies for protecting these resources within the network. This paper presents a comprehensive set of technical justifications to explain the manners in which Windows Vista and Windows Server 2008 collectively address a modern set of Commercial Grade Operating System Security Functional Requirements.

Version 2.0 of this paper also includes, in its Appendix B, the specific technical justifications for addressing certain additional interesting security functional requirements in the areas of Security Architecture, Access Control Polices, and Information Protection.



The information contained in this document represents the current view of Microsoft Corporation on the issues discussed as of the date of publication. Because Microsoft must respond to changing market conditions, it should not be interpreted to be a commitment on the part of Microsoft, and Microsoft cannot guarantee the accuracy of any information presented after the date of publication.

This White Paper is for informational purposes only. MICROSOFT MAKES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION IN THIS DOCUMENT.

Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

© 2009 Microsoft Corporation. All rights reserved.

Microsoft, MS-DOS, Bitlocker, Windows, Windows Server, and Windows Vista are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

All other trademarks are property of their respective owners.

# Introduction

An operating system (OS) is critical for enabling general information technology (IT) services over a network for users. Because of the fundamental resources being consumed by the IT services, the OS must enforce appropriate security policies for protecting these resources within the network.

Within the framework of [Common Criteria (CC)](http://www.commoncriteriaportal.org/thecc.html) ([ISO/IEC 15408 “Information technology -- Security techniques -- Evaluation criteria for IT security”](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40612)), the “[Controlled Access Protection Profile (CAPP)](http://www.niap-ccevs.org/cc-scheme/pp/id/pp_os_ca_v1.d)” traditionally has provided the most commonly used set of security functional requirements for an operating system. Many commercial OS products have been successfully evaluated by 3rd parties for compliance with the [CAPP](http://www.niap-ccevs.org/cc-scheme/pp/id/pp_os_ca_v1.d) under the [CC](http://www.commoncriteriaportal.org/thecc.html) ([ISO/IEC 15408](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40612)). However, the [CAPP](http://www.niap-ccevs.org/cc-scheme/pp/id/pp_os_ca_v1.d) is in the process of being made obsolete by its author/maintainer. Modern commercial OS products need to address newer security requirements identified in modern IT service operational environments. These newer security requirements do not arise from the security objectives that originally underlay the [CAPP](http://www.niap-ccevs.org/cc-scheme/pp/id/pp_os_ca_v1.d). The on-going development of the following two protection profiles is intended to replace the [CAPP](http://www.niap-ccevs.org/cc-scheme/pp/id/pp_os_ca_v1.d) for the more modern OS products.

* “[US Government Protection Profile for Single-Level Operating Systems in a Medium Robustness Environments](http://www.niap-ccevs.org/cc-scheme/pp/id/pp_os_sl_mr2.0_v1.91)”;
* “[US Government Protection Profile for Multi-Level Operating Systems in a Medium Robustness Environments](http://www.niap-ccevs.org/cc-scheme/pp/id/pp_os_ml_mr2.0_v1.91)”.

However, these two newer protection profiles are not yet stable enough for an OS product to be evaluated for compliance with either protection profile. To date, there has not been an announcement on the [official Common Criteria public portal](http://www.commoncriteriaportal.org/products_OS.html#OS) that an OS product has been evaluated for compliance with either protection profile under the [CC](http://www.commoncriteriaportal.org/thecc.html) ([ISO/IEC 15408](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40612)).

[NIST SP 800-53 “Recommended Security Controls for Federal Information Systems”](http://csrc.nist.gov/publications/nistpubs/800-53-Rev2/sp800-53-rev2-final.pdf) is another good source of security functional requirements. However, [NIST SP 800-53](http://csrc.nist.gov/publications/nistpubs/800-53-Rev2/sp800-53-rev2-final.pdf) does not provide specific guidelines for determining which particular recommended security control elements are meant to be addressed by an operating system product.

During 2007, the author was in a fortunate position, where a number of very qualified IT security professionals were willing to share their thoughts on the security functional requirements that a modern commercial grade OS is expected to meet, in today’s general IT service operational environments. Through private communications with these IT security professionals, a Commercial Grade Operating System Security Functional Requirement Set was resulted at the end of 2007. The details of this Commercial Grade Operating System Security Functional Requirement Set are listed in the following section of this paper.

* “A Commercial Grade Operating System Security Functional Requirement Set”.

This paper presents a comprehensive set of technical justifications to explain the manner in which Windows Vista and Windows Server 2008 (collectively known as the Windows OS in this paper) address the security functional requirements of this “Commercial Grade Operating System Security Functional Requirement Set”.

While the technical justifications for addressing the individual requirements are presented in the corresponding sections of this paper, a summary of the Windows OS’s compliance with the requirements is presented in the following section.

* “Summary of Windows’ Compliance with the Commercial Grade Operating System Security Functional Requirement Set”.

The author would like to thank the following individuals, especially David Cross, for their valuable suggestions that helped to improve the earlier drafts of this paper.

Mark Russinovich, Douglas MacIver, Eric Perlin, Jon Schwartz, Scott Field, Richard B. Ward, Glenn Pittaway, Steve Lipner.

# A Commercial Grade Operating System Security Functional Requirement Set

At the highest level, the Commercial Grade OS Requirement Set consists of 8 areas of security functionality. Each area may have one or more subareas. These areas of security functionality are:

* Security Architecture:
  + Separation, Isolation, and Least Privilege;
  + Distributed Architectures;
* Access Control Polices:
  + Discretionary Access Control Policy;
  + Mandatory Integrity Control Policy;
* Identification and Authentication:
  + User Identification/Authentication, Attributes, Roles, and Re-Authentication;
  + User Interface Security;
* Cryptographic Support:
  + Operations;
  + Capabilities;
  + Randomization;
  + Cryptographic Key Management;
  + Cryptographic Testing;
* Information Protection:
  + Residual Information;
  + Resource Control;
  + Trusted Initialization~~, Self Testing and Recovery~~;
* Import/Export of Data:
  + Exported Data;
  + Imported Data;
  + Trusted Channels;
* Revocation:
  + Access Revocation;
* Auditing:
  + Audit Collection;
  + Audit Storage.

Individual requirements are specified as security functional requirements under a specific subarea of security functionality. Where applicable, the corresponding management and/or auditing aspects of a security functional requirement are also specified as a management requirement and/or audit requirement respectively in each subarea.

|  |  |  |  |
| --- | --- | --- | --- |
| **Security Architecture: Separation, Isolation, and Least Privilege**  Separation, Isolation and Least privilege are the fundamental principles of security. Without them, all other claims of “security” become meaningless.    Separation is the characteristic of providing each subject with its own execution and data space; any sharing of data and resources with other subjects is mediated by the security functions and is based upon some rule set.  Isolation is the characteristic of ensuring that resources (e.g., processes, data containers, system objects) are protected from unauthorized access. All sharing of information between resources (possibly even the existence of resources) is controlled by the system security policy.  Least privilege is the characteristic whereby an entity (e.g. subject) has only the minimum privileges (authorizations, permissions, etc.) required to function and has them only when it needs them. This helps ensure that, should something go amiss, the extent of resulting damage would be minimal.  The implementation of these security principles are fundamental aspects of what is commonly referred to as a reference monitor. The characteristics of the implementation of these security principles, relative to the operating system (OS) security, are:   1. that it is always invoked (i.e., it cannot be bypassed or subverted and the appropriate security policies are applied to all actions); 2. that it is tamper proof (i.e., it cannot be modified or interfered with); and 3. that the implementation is conceptually simple enough to allow for thorough analysis and understanding.   Because they are characteristics of the operating system, rather than security functions per se, they cannot be readily observed or demonstrated through testing alone. Analysis is necessary to verify conformance with these requirements. | | | |
|  | **Functional Requirements** | | |
|  |  | 1.1.1.1 | The OS shall protect security-relevant data from unauthorized access.  *Application Note: The OS security relevant data includes all data on which the OS relies for its correct operation and enforcement of its security policies.* |
|  |  | 1.1.1.2 | The OS shall ensure that all security policies are enforced before each security function is allowed to proceed.  *Application note: This requirement captures the fundamental security notion of “Always invoked”.* |
|  |  | 1.1.1.3 | The OS shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.  *Application note: This requirement captures the fundamental security notion of “Tamper proof”.* |
|  |  | 1.1.1.4 | The OS shall enforce separation between the security domains of subjects.  *Application note: This requirement captures the fundamental security notion of “Separation”.* |
|  |  | 1.1.1.5 | The OS shall make effective use of hardware provided security features. |
|  | **Management Requirements** | | |
|  |  | 1.1.2.x | None. |
|  | **Audit Requirements** | | |
|  |  | 1.1.3.x | None. |
|  | | | |
| **Security Architecture: Distributed Architectures**  An OS, that is distributed, presents additional security architectural concerns of synchronization of security-relevant data and protection of this data while in transit between distributed parts of the OS. | | | |
|  | **Functional Requirements** | | |
|  |  | 1.2.1.1 | The OS shall ensure that security-relevant data is consistent between parts of the OS by providing a mechanism to bring inconsistent data into a consistent state in a timely manner.  *Application Note: In a distributed operating system, synchronization of security-relevant data (such as access permissions, user authorizations, quota usage, last successful and unsuccessful logon attempts) is necessary to ensure the correct enforcement of the system security policies. However, it is impossible to achieve complete, constant consistency of security-relevant data that is distributed to remote portions of an OS because distributed portions of the OS may be active at different times or disconnected from one another. This requirement attempts to address this situation in a practical manner by acknowledging that there will be security-relevant data inconsistencies but that they will be corrected without undue delay.* |
|  |  | 1.2.1.2 | The OS shall protect security-relevant data from disclosure while being transmitted to a remote part of the OS through the use of OS provided cryptographic services. |
|  |  | 1.2.1.3 | The OS shall detect modification and insertion of security-relevant received from a remote part of the OS through the use of OS provided cryptographic services. |
|  |  | 1.2.1.4 | Upon detection of modification and insertion of security-relevant received from a remote part of the OS, the OS shall reject the data. |
|  |  | 1.2.1.5 | The OS shall provide a means for secure remote administration using OS provided cryptographic services. |
|  |  | 1.2.1.6 | The OS shall authenticate remote parts of the OS using OS provided cryptographic services. |
|  | **Management Requirements** | | |
|  |  | 1.2.2.1 | The OS shall provide the ability for an authorized administrator to remotely manage the OS. |
|  | **Audit Requirements** | | |
|  |  | 1.2.3.1 | The OS shall provide the ability to audit when modification or insertion of security-relevant data received from a remote part of the OS has been detected. |
|  | | | |
| **Access Control Polices: Discretionary Access Control Policy**  The Discretionary Access Control (DAC) is a security policy that enforces rules defining how user code and data may be shared among untrusted users. These rules are typically based upon the identities of users and are typically defined by the owners of the data. | | | |
|  | **Functional Requirements** | | |
|  |  | 2.1.1.1 | The OS shall enforce a Discretionary Access Control policy on all subjects, all named objects and all access operations among them. |
|  |  | 2.1.1.2 | The OS shall associate Discretionary Access Control security attributes with each subject and named object. |
|  |  | 2.1.1.3 | The OS shall enforce the Discretionary Access Control policy on named objects based on the specific types of the subject and object security attributes.  The Commercial Grade OS Requirement Set requires the following types of subject and object security attributes:   1. The authorized user identity and group membership(s) associated with a subject; 2. The {identity, access operations} pairs associated with a named object. |
|  |  | 2.1.1.4 | The OS shall provide restrictive default values for object security attributes that are used to enforce the Discretionary Access Control policy. |
|  |  | 2.1.1.5 | The OS shall ensure that only valid values are accepted for an object’s Discretionary Access Control security attributes. |
|  |  | 2.1.1.6 | The Discretionary Access Control policy mechanism shall provide that named objects are protected from unauthorized access according to a specific set of ordered rules.  The Commercial Grade OS Requirement Set requires the following ordered rules.   1. If the requested mode of access is explicitly denied to the requesting user, deny access; 2. If the requested mode of access is permitted to the requesting user, permit access; 3. If the requested mode of access is explicitly denied to every group of which the requesting user is a member, deny access; 4. If the requested mode of access is permitted to any group of which the requesting user is a member, grant access; 5. Else, deny access. |
|  | **Management Requirements** | | |
|  |  | 2.1.2.1 | The OS shall allow only authorized administrators, object owners, and users with the DAC change authorization the ability to change the access permission associated with a named object. |
|  |  | 2.1.2.2 | The OS shall allow only authorized administrators the ability to change object ownership. |
|  |  | 2.1.2.3 | The OS shall provide only authorized administrators the ability to specify alternative initial values for default Discretionary Access Control object security attributes. |
|  | **Audit Requirements** | | |
|  |  | 2.1.3.1 | The OS shall provide the ability to audit all Discretionary Access Control policy decisions. |
|  |  | 2.1.3.2 | The OS shall provide the ability to audit all changes to an object’s Discretionary Access Control security attributes. |
|  |  | 2.1.3.3 | The OS shall provide the ability to audit all modifications to default Discretionary Access Control object security attribute values. |
|  | | | |
| **Access Control Polices: Mandatory Integrity Control Policy**  Mandatory Integrity Control (MIC) is a rule-based policy that addresses the integrity or trustworthiness of information. It is mandatory in the sense that an authorized administrator sets up the rules that determine how subjects may modify the data contained within objects. The owner of the object cannot specify authorizations to obviate these rules. | | | |
|  | **Functional Requirements** | | |
|  |  | 2.2.1.1 | The OS shall enforce a Mandatory Integrity Control policy on untrusted subjects, named objects, and network interfaces and all operations that cause information to flow among them. |
|  |  | 2.2.1.2 | The OS shall associate an integrity label with each subject, named object, and network interface that accurately represents its integrity level. |
|  |  | 2.2.1.3 | The OS shall enforce the Mandatory Integrity Control policy based on the specific types of the subject and object integrity attributes.  The Commercial Grade OS Requirement Set requires the following types of subject and object integrity attributes:   1. integrity labels of subjects; 2. integrity labels of objects; 3. integrity labels consisting of at least 2 definable hierarchical levels or a set of at least 2 definable non-hierarchical categories; 4. any additional integrity attributes (e.g. privileges). |
|  |  | 2.2.1.4 | The OS shall enforce specific relationships for any two valid integrity labels.  The Commercial Grade OS Requirement Set requires one of the following three selections of specific relationships for any two valid integrity labels.   1. For hierarchical integrity policy schemes: there exists an ordering function that, given two valid integrity labels, determines if the integrity labels are equal or if one integrity label is greater than the other; 2. For non-hierarchical integrity attributes schemes, integrity labels are composed of a set of distinct and non-comparable integrity attributes; 3. For integrity policy schemes that include both hierarchical and non-hierarchical components:    1. There exists an ordering function that, given two valid integrity labels, determines if the integrity labels are equal or if one integrity label dominates the other.    2. There exists a “least upper bound” in the set of integrity labels, such that, given any two valid integrity labels, there is a valid integrity label that dominates the two valid integrity labels.    3. There exists a “greatest lower bound” in the set of integrity labels, such that, given any two valid integrity labels, there is a valid integrity label that is dominated by the two valid integrity labels. |
|  |  | 2.2.1.5 | The OS shall permit an information flow among subjects and objects based on a specific set of rules.  The Commercial Grade OS Requirement Set requires one of the following two selections of specific sets of rules for information flows among subjects and objects.   1. For hierarchical integrity attributes schemes:    1. If the integrity label of the subject is greater than or equal to the integrity label of the object, then a write (the flow of information from the subject to the object) is permitted;    2. If the integrity label of the object is greater than or equal to the integrity label of the subject, then a read (the flow of information from the object to the subject) is permitted;    3. If the information flow is between objects, the integrity label of the source object must be greater than or equal to the integrity label of the destination object. 2. For non-hierarchical integrity attributes schemes:    1. If the integrity label of the object is equal to the integrity label of the subject, then the flow of information between the object and the subject is permitted;    2. If the integrity label of the subject dominates the integrity label of the object (i.e. the object’s label is a subset of the subject’s label), then a write (the flow of information from the subject to the object) is permitted;    3. If the integrity label of the object dominates the integrity label of the subject (i.e. the subject’s label is subset of the object’s label), then a read (the flow of information from the object to the subject) is permitted;    4. If the integrity label of the subject and the integrity label of the object have no relationship (i.e. one is not a subset of the other) then they are non-comparable and as a consequence information flow is not allowed. |
|  |  | 2.2.1.6 | The OS shall enforce that all definition of integrity labels are unique. |
|  | **Management Requirements** | | |
|  |  | 2.2.2.1 | The OS shall provide the ability to set the system-wide definition of integrity labels to authorized administrators. |
|  |  | 2.2.2.2 | The OS shall provide authorized administrators the ability to change integrity labels of subjects and objects. |
|  | **Audit Requirements** | | |
|  |  | 2.2.3.1 | The OS shall provide the ability to audit all Mandatory Integrity Control policy decisions. |
|  |  | 2.2.3.2 | The OS shall provide the ability to audit the setting and changing of system-wide integrity label definitions. |
|  |  | 2.2.3.3 | The OS shall provide the ability to audit the changing of subject and object integrity labels. |
|  | | | |
| **Identification and Authentication: User Identification/Authentication, Attributes, Roles, and Re-Authentication**  When a user claims to have a specific identity, authentication is the means by which the OS confirms that claimed identity. The OS also imposes requirements concerning characteristics about the authentication data that is used to confirm the claimed identity, the roles which that user may assume (upon successful authentication of the claimed identity) and the rules for the re-authentication of the claimed identity. | | | |
|  | **Functional Requirements** | | |
|  |  | 3.1.1.1 | The OS shall require each user to be uniquely identified and successfully authenticated by means of a password before allowing any actions on behalf of that user. |
|  |  | 3.1.1.2 | The OS shall be able to support passwords up to 32 characters in length, consisting of any combination of upper and lower case letters, numbers, and punctuations. |
|  |  | 3.1.1.3 | The OS authentication mechanism shall provide a specific set of capabilities.  The Commercial Grade OS Requirement Set requires the following specific set of capabilities:   1. for all administrator accounts, a delay such that there can be no more than ten attempts per minute; 2. for all other accounts, a delay such that there can be no more than twenty attempts per minute. |
|  |  | 3.1.1.4 | The OS shall provide only obscured feedback to the user while the authentication is in progress. |
|  |  | 3.1.1.5 | The OS shall detect when an authorized administrator specified positive integer of consecutive unsuccessful authentication attempts occur related to any authorized user authentication process. |
|  |  | 3.1.1.6 | The OS shall perform a specific set of actions when the defined number of consecutive unsuccessful authentication attempts specified in “3.1.1.5” has been detected.  The Commercial Grade OS Requirement Set requires the following specific set of actions being performed by the OS:   1. for all administrator accounts, disable the account for an authorized administrator configurable time period; 2. for all other accounts, disable the user account until it is re-enabled by an authorized administrator; 3. for all disable accounts, respond with an “account disabled” message without attempting any type of authentication. |
|  |  | 3.1.1.7 | The OS shall maintain a specific list of security attributes belonging to individual users.  The Commercial Grade OS Requirement Set requires the following specific list of security attributes belonging to individual users being maintained by the OS:   1. unique user identity; 2. group memberships; 3. authentication data; 4. any other security-relevant authorizations or attributes (e.g. roles). |
|  |  | 3.1.1.8 | The OS shall associate a specific list of user security attributes with subjects acting on behalf of that user.  The Commercial Grade OS Requirement Set requires the following specific list of user security attributes for associating with subjects acting on behalf of that user:   1. the unique user identity; 2. any group identity or identities; 3. any other security-relevant authorizations or attributes (e.g. roles). |
|  |  | 3.1.1.9 | The OS shall protect authentication data from disclosure through the use of operating system provided cryptographic services. |
|  |  | 3.1.1.10 | The OS shall provide an administrator role that is separate from untrusted users. |
|  |  | 3.1.1.11 | The OS shall provide the authorized administrator the ability to set user attributes. |
|  |  | 3.1.1.12 | The OS shall re-authenticate the user when changing authentication data. |
|  |  | 3.1.1.13 | The OS shall provide a mechanism to verify that each user password when set meets authorized administrator configurable password characteristics. |
|  |  | 3.1.1.14 | The OS shall automatically disable a user account when the current time has passed the authorized administrator specified time interval in which the user account has not been logged on successfully since the user account’s last successful logon time. |
|  | **Management Requirements** | | |
|  |  | 3.1.2.1 | The OS shall allow only authorized administrators the ability to create and manage user accounts. |
|  |  | 3.1.2.2 | The OS shall provide authorized administrators the ability to specify a time interval in which the user account has not been logged on successfully so that a user account is automatically disabled after the time interval has elapsed. |
|  |  | 3.1.2.3 | The OS shall allow only authorized administrators the ability to initially set and modify user security attributes (other than authentication data). |
|  |  | 3.1.2.4 | The OS shall allow only authorized administrators the ability to initialize user authentication data. |
|  |  | 3.1.2.5 | The OS shall restrict the ability to modify authentication data to authorized administrators and users authorized to modify their own authentication data. |
|  |  | 3.1.2.6 | The OS shall allow only authorized administrators the ability to initialize and modify authentication mechanism attributes. |
|  |  | 3.1.2.7 | The OS shall provide authorized administrators the ability to specify and configure mandatory password composition. |
|  | **Audit Requirements** | | |
|  |  | 3.1.3.1 | The OS shall provide the ability to audit the creation and management of user accounts. |
|  |  | 3.1.3.2 | The OS shall provide the ability to audit the initialization and modification of user security attributes. |
|  |  | 3.1.3.3 | The OS shall provide the ability to audit the initialization and modification of authentication mechanism attributes. |
|  |  | 3.1.3.4 | The OS shall provide the ability to audit when a user’s consecutive unsuccessful authentication attempts meets or exceeds the administrator-configured positive integer. |
|  |  | 3.1.3.5 | The OS shall provide the ability to audit all user attempts to identify and authenticate to the system. |
|  | | | |
| **Identification and Authentication: User Interface Security**  The OS also needs to ensure that, once a user has been successfully authenticated, the operational association established with that user is maintained. | | | |
|  | **Functional Requirements** | | |
|  |  | 3.2.1.1 | Before establishing an interactive session, the OS shall display an authorized administrator specified advisory notice. |
|  |  | 3.2.1.2 | The OS shall allow user-initiated locking of an interactive session by performing a specific set of actions.  The Commercial Grade OS Requirement Set requires the following specific set of actions being performed by the OS:   1. Clearing or overwriting display devices, making the current contents unreadable; 2. Disabling any activity from the user’s data access/display devices other than unlocking the interactive session. |
|  |  | 3.2.1.3 | The OS shall require a specific set of actions from users and administrators.  The Commercial Grade OS Requirement Set requires the following specific set of actions from users and administrators:   1. A user (needs) to re-authenticate to unlock an interactive session; 2. An administrator (needs) to authenticate before the system security functions unlock and automatically terminate a user interactive session. |
|  |  | 3.2.1.4 | The OS shall enforce an authorized administrator specified maximum number of concurrent interactive sessions per user. |
|  |  | 3.2.1.5 | The OS shall lock an interactive session after an authorized administrator specified time interval of user inactivity by performing a specific set of actions.  The Commercial Grade OS Requirement Set requires the following specific set of actions being performed by the OS:   1. Clearing or overwriting display devices, making the current contents unreadable; 2. Disabling any interactive user activity from that display devices other than unlocking the interactive session. |
|  |  | 3.2.1.6 | The OS shall provide the ability to deny interactive session establishment based on time and day. |
|  |  | 3.2.1.7 | Upon successful interactive session establishment, the OS shall display to the authorized user a specific set of user access history information elements.  The Commercial Grade OS Requirement Set requires the following specific set of user access history information elements to be displayed to the authorized user:   1. Date, time, and location of that user’s last successful interactive session establishment; 2. The number of unsuccessful (interactive session establishment) authentication attempts for that user since last successful interactive session establishment. |
|  |  | 3.2.1.8 | The OS shall not erase the user access history information elements from the authorized user interface without giving the authorized user the opportunity to review the information elements. |
|  | **Management Requirements** | | |
|  |  | 3.2.2.1 | The OS shall provide authorized administrators with the ability to specify an advisory notice. |
|  |  | 3.2.2.2 | The OS shall provide authorized administrators with the ability to terminate an interactive session. |
|  |  | 3.2.2.3 | The OS shall provide authorized administrators with the ability to specify the number of concurrent interactive sessions allowed per user. |
|  |  | 3.2.2.4 | The OS shall provide authorized administrators with the ability to deny interactive session establishment based on system parameters specified in “3.2.1.6”. |
|  |  | 3.2.2.5 | The OS shall provide authorized administrators with the ability to specify a time interval of session inactivity after which an (inactive) interactive session is locked. |
|  | **Audit Requirements** | | |
|  |  | 3.2.3.1 | The OS shall provide the ability to audit all user attempts to re-authenticate to the system (for the interactive session unlocking purpose). |
|  |  | 3.2.3.2 | The OS shall provide the ability to audit all administrator attempts to terminate a user’s locked session. |
|  |  | 3.2.3.3 | The OS shall provide the ability to audit any attempt to exceed the maximum number of concurrent interactive sessions by a user. |
|  |  | 3.2.3.4 | The OS shall provide the ability to audit all user attempts that violate the restrictions specified in “3.2.1.6”. |
|  | | | |
| **Cryptographic Support: Operations**  The OS needs to provide cryptographic services with an appropriate commercial grade qualification. | | | |
|  | **Functional Requirements** | | |
|  |  | 4.1.1.1 | The OS shall provide a specific list of cryptographic services to applications.  The Commercial Grade OS Requirement Set requires the following list of cryptographic services to be provided to applications:   1. Encryption/Decryption; 2. Digital Signature; 3. Hashing; 4. Cryptographic Key Agreement; 5. Randomization; 6. Key Management. |
|  | **Management Requirements** | | |
|  |  | 4.1.2.x | None. |
|  | **Audit Requirements** | | |
|  |  | 4.1.3.x | None. |
|  | | | |
| **Cryptographic Support: Capabilities**  Arguably, an appropriate commercial grade qualification can be defined through the FIPS approval criteria, implying also ISO/IEC 19790 compliance. FIPS approval is based on [FIPS 140-2 “Security Requirements for Cryptographic Modules”](http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf). [ISO/IEC 19790 “Information technology -- Security techniques -- Security requirements for cryptographic modules”](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=33928) is derived from FIPS 140-2. In the US and Canada, an official FIPS 140-2 validation has to be achieved through the “[Derived Test Requirements [DTRs] for FIPS 140-2, Security Requirements for Cryptographic Modules](http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/fips1402DTR.pdf)” being conducted by an [accredited Cryptographic Module Testing (CMT) laboratory](http://csrc.nist.gov/groups/STM/testing_labs/index.html). [ISO/IEC 24759 “Information technology -- Security techniques -- Test requirements for cryptographic modules”](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=41529) is also derived from the DTRs for FIPS-140-2. | | | |
|  | **Functional Requirements** | | |
|  |  | 4.2.1.1 | The OS shall provide FIPS-approved (i.e. implying also ISO/IEC compliant) cryptographic algorithms and modes of operations, implemented in a crypto module that is FIPS 140-2 Security Level 2 validated (i.e. implying also ISO/IEC 19790 compliant).  The Commercial Grade OS Requirement Set requires the evidence of the conformance with the following list of FIPS 140-2 Security Level 2 validation areas in the FIPS 140-2 validated crypto module:   * Cryptographic Module Specification; * Cryptographic Module Ports and Interfaces; * Roles, Services and Authentication; * Finite State Model; * Cryptographic Key Management; * Self-Tests; * Design Assurance.   *Application Note: The conformance with the remaining FIPS 140-2 Level 2 validation areas of Physical Security, Operational Environment, EMI/EMC, and Mitigation of Other Attacks is not necessary.* |
|  |  | 4.2.1.2 | A specific list of algorithms shall be provided for the corresponding cryptographic services.  The Commercial Grade OS Requirement Set requires the following list of algorithms.   1. For encryption/decryption, either    1. Triple Data Encryption Algorithm (TDEA) used in FIPS-approved modes of operation and cryptographic key size of at least 168 bits (three independent keys);    2. Advanced Encryption Standard (AES) used in FIPS-approved modes of operation and cryptographic key size of at least 128 bits; 2. For digital signature, either    1. Digital Signature Algorithm (DSA) with a key size (modulus) of at least 1028 bits;    2. RSA Digital Signature Algorithm (rDSA) with a key size of at least 2048 bits;    3. Elliptic Curve Digital Signature Algorithm (ECDSA) using only the NIST-curves with a key size of at least 256 bits, using NIST P curves; 3. For hashing, either    1. SHA-1;    2. SHA-256;    3. SHA-512; 4. For key agreement, either    1. Finite Field-based key agreement algorithm and cryptographic key sizes (modulus) of at least 2048 bits as specified in [FIPS 140-2 Annex D];    2. Elliptic Curve-based key agreement algorithm and cryptographic key size of at least 256 bits as specified in [FIPS 140-2 Annex D]. |
|  | **Management Requirements** | | |
|  |  | 4.2.2.1 | The OS shall provide the authorized administrators the ability to select (when more than one algorithm is available) the appropriate algorithm for encryption/decryption, digital signature, hashing, and key agreement. |
|  | **Audit Requirements** | | |
|  |  | 4.2.3.x | None. |
|  | | | |
| **Cryptographic Support: Randomization** | | | |
|  | **Functional Requirements** | | |
|  |  | 4.3.1.1 | The OS crypto module shall provide Random Number Generation (RNG) services in accordance with a FIPS-Approved RNG listed in [FIPS 140-2 Annex C] composed of a specific method.  The Commercial Grade OS Requirement Set requires the selection of one of the following RNG method that complies with the tests specified in [NIST SP 800-90].   1. At least one independent hardware-generated input (noise source) combined with a FIPS-approved cryptographic hashing function; 2. At least one independent software-generated input (noise source) combined with a FIPS-approved cryptographic hashing function; 3. A combination of at least one independent hardware-generated input (noise source) combined with a mixing function and at least one independent software-generated input (noise source) combined with a FIPS-approved cryptographic hashing function. |
|  | **Management Requirements** | | |
|  |  | 4.3.2.x | None. |
|  | **Audit Requirements** | | |
|  |  | 4.3.3.x | None. |
|  | | | |
| **Cryptographic Support: Cryptographic Key Management** | | | |
|  | **Functional Requirements** | | |
|  |  | 4.4.1.1 | The OS crypto module shall generate symmetric cryptographic keys using a random number generator as specified in “4.3.1.1”. |
|  |  | 4.4.1.2 | The OS crypto module shall generate asymmetric cryptographic keys using a domain parameter generator and a random number generator as specified in “4.3.1.1”. |
|  |  | 4.4.1.3 | The OS crypto module shall generate cryptographic keys such that generated key strength shall be equivalent to, or greater than, a symmetric key strength of 128 bits using conservative estimates as specified in [NIST SP 800-57 Section 5.6.1]. |
|  |  | 4.4.1.4 | The OS crypto module shall perform key entry and output in accordance with the Level 1 Key Entry and Output requirements in [FIPS 140-2], “Security Requirements for Cryptographic Modules”. |
|  |  | 4.4.1.5 | The OS crypto module shall provide the ability to protect secret keys, private keys, and critical cryptographic security parameters from unauthorized disclosure, modification, and substitution. |
|  |  | 4.4.1.6 | The OS crypto module shall provide the ability to protect public keys from unauthorized modification, and substitution. |
|  |  | 4.4.1.7 | The OS crypto module shall provide the ability to correctly associate stored cryptographic keys (secret, private, or public) with the entity (e.g. person, group, or subject) to which the key is assigned. |
|  |  | 4.4.1.8 | The OS crypto module shall destroy cryptographic keys in accordance with a specific cryptographic key zeroization method.  The Commercial Grade OS Requirement Set requires the specific cryptographic key zeroization method to meet the following requirements.   1. Key zeroization requirements in [FIPS PUB 140-2], “Security Requirements for Cryptographic Modules.” 2. Zeroization of all plaintext cryptographic keys and all other critical cryptographic security parameters shall be immediate and complete. 3. Each intermediate storage area for plaintext cryptographic key /critical cryptographic security parameter (i.e. any storage, such as memory buffers, that is included in the path of such data) shall be zeroized. 4. Zeroization shall be executed by one of the following two selections:    1. For non-volatile memories other than EEPROM and Flash, the zeroization shall be executed by overwriting three or more times using a different alternating data pattern each time;    2. For volatile memories and non-volatile EEPROM and Flash memories, the zeroization shall be executed by a single direct overwrite consisting of a pseudo random pattern, followed by a read-verify. |
|  | **Management Requirements** | | |
|  |  | 4.4.2.x | None. |
|  | **Audit Requirements** | | |
|  |  | 4.4.3.1 | The OS shall provide the ability to audit attempts to violate the restrictions specified in “4.4.1.5”. |
|  |  | 4.4.3.2 | The OS shall provide the ability to audit attempts to violate the restrictions specified in “4.4.1.6”. |
|  | | | |
| **Cryptographic Support: Cryptographic Testing**  A demonstration of the correct operations of the OS provided cryptographic services is desired. | | | |
|  | **Functional Requirements** | | |
|  |  | 4.5.1.1 | The OS crypto module shall run a suite of self-tests to demonstrate the correct operation of the cryptographic functions in accordance with [FIPS 140-2 Section 4.9] during specific occasions.  The Commercial Grade OS Requirement Set requires the following specific occasions to run the self tests.   1. The initial start-up (on power on); 2. At the request of an authorized administrator (on demand); 3. Under various conditions defined in [FIPS 140-2 Section 4.9]; 4. Periodically (at least once a day). |
|  |  | 4.5.1.2 | If any of the cryptographic self-tests fail, the OS shall react as required by [FIPS 140-2 Section 4.9] for failing a self-test. |
|  |  | 4.5.1.3 | The OS crypto module shall run a set of specific additional RNG tests.  The Commercial Grade OS Requirement Set requires the following specific additional RNG tests.   1. All known answer RNG tests of [NIST SP 800-90] upon demand and upon power on; 2. The noise sources shall undergo a test at startup to ensure that constant data is not being produced. |
|  | **Management Requirements** | | |
|  |  | 4.5.2.1 | The OS shall provide the ability for an authorized administrator to run a suite of crypto module self-tests and RNG tests. |
|  | **Audit Requirements** | | |
|  |  | 4.5.3.1 | The OS shall audit failure of any crypto module self-tests and RNG tests. |
|  | | | |
| **Information Protection: Residual Information**  The OS obviously needs to support the reuse of resources, such as memory areas (e.g. buffers and pages) and secondary storage areas (e.g. disks) that contain user data. Once a user no longer needs a given storage resource to house data, that resource must be appropriately handled to ensure that the security policies remain enforced. | | | |
|  | **Functional Requirements** | | |
|  |  | 5.1.1.1 | The OS shall ensure that any previous information content of a resource is made unavailable upon either the allocation of the resource to or the de-allocation of the resource from all non cryptographic objects. |
|  |  | 5.1.1.2 | Objects associated with cryptographic keys and critical cryptographic security parameters shall be destroyed as described in “4.4.1.8”. |
|  | **Management Requirements** | | |
|  |  | 5.1.2.x | None. |
|  | **Audit Requirement** | | |
|  |  | 5.1.3.x | None. |
|  | | | |
| **Information Protection: Resource Control** | | | |
|  | **Functional Requirements** | | |
|  |  | 5.2.1.1 | The OS shall enforce maximum quotas on the portion of shared persistent storage that individual authorized users can use. |
|  | **Management Requirement** | | |
|  |  | 5.2.2.1 | The OS shall allow only authorized administrators the ability to set maximum quotas on shared persistent storage. |
|  | **Audit Requirements** | | |
|  |  | 5.2.3.1 | The OS shall provide the ability to audit any request by a user that attempts to exceed the authorized administrator defined shared persistent storage quota. |
|  |  | 5.2.3.2 | The OS shall provide the ability to audit the setting of maximum quotas on shared persistent storage. |
|  | | | |
| **Information Protection: Trusted Initialization~~, Self Testing and Recovery~~** | | | |
|  | **Functional Requirements** | | |
|  |  | ~~5.3.1.1~~ | ~~The OS shall run a suite of self tests to demonstrate the correct operation of the OS security functions during specific occasions.~~ |
|  |  | ~~5.3.1.2~~ | ~~After a failure of a self test, the OS shall ensure a maintenance mode where the ability to return the OS to a secure state is provided.~~ |
|  |  | 5.3.1.3 | The OS shall verify during initial startup the integrity of executable code that implements access control and cryptographic functionality through the use of the OS system provided cryptographic services. |
|  | **Management Requirements** | | |
|  |  | ~~5.3.2.1~~ | ~~The OS shall provide authorized administrators the ability to run the OS self tests.~~ |
|  |  | ~~5.3.2.2~~ | ~~The OS shall provide authorized administrators the ability to define the frequency at which self tests are automatically run.~~ |
|  |  | 5.3.2.x | None. |
|  | **Audit Requirements** | | |
|  |  | ~~5.3.3.1~~ | ~~The OS shall provide the ability to audit any self test failures.~~ |
|  |  | ~~5.3.3.2~~ | ~~The OS shall provide the ability to audit when authorized administrators initiate the OS self tests.~~ |
|  |  | 5.3.3.3 | The OS shall provide the ability to audit the failure of the (executable code) integrity verification. |
|  | | | |
| **Import/Export of Data: Exported Data**  The OS needs to ensure the binding of security attributes to named objects and protection of named objects when it is transmitted or transferred from the OS. | | | |
|  | **Functional Requirements** | | |
|  |  | 6.1.1.1 | The OS shall ensure that security attributes on named objects, when exported to removable media, are associated with the object. |
|  | **Management Requirements** | | |
|  |  | 6.1.2.x | None. |
|  | **Audit Requirements** | | |
|  |  | 6.1.3.1 | The OS shall provide the ability to audit the exportation of named objects to removable media. |
|  | | | |
| **Import/Export of Data: Imported Data**  The OS needs to handle imported user data security attributes and the protection against disclosure of data that is attempted for an import to the OS. | | | |
|  | **Functional Requirements** | | |
|  |  | 6.2.1.1 | The OS shall enforce a specific set of rules when importing user data.  The Commercial Grade OS Requirement Set requires the following rules.   1. Import with Attributes    1. validate the security attributes.    2. if validation succeeds, the OS shall       1. ensure that the validated security attributes are associated with the imported user data, and       2. ensure that all applicable access control policies are enforced by the act of importing the user data.    3. if validation fails, the OS shall associate security attributes with the imported data according to the following authorized administrator specified setting selection:       1. the DAC and MIC security attributes of the importing subject;       2. deleting the data. 2. Import without Attributes    1. associate security attributes with the imported user data according to the following authorized administrator specified setting selection:       1. the DAC and MIC security attributes of the importing subject;       2. deleting the data. |
|  | **Management Requirements** | | |
|  |  | 6.2.2.1 | The OS shall provide authorized administrators with the ability to select the technique for handling imported data with attributes that cannot be validated. |
|  | **Audit Requirements** | | |
|  |  | 6.2.3.1 | The OS shall provide the ability to audit the failure of security attribute validation. |
|  |  | 6.2.3.2 | The OS shall provide the ability to audit the administrator selected configuration for handling imported data with attributes that cannot be validated. |
|  | | | |
| **Import/Export of Data: Trusted Channels**  The protection of specific information that is exchanged between an interactive user and the OS is critical. | | | |
|  | **Functional Requirements** | | |
|  |  | 6.3.1.1 | The OS shall provide a communication path between itself and users that is logically distinct from other communication paths and provides assured identification of the OS to the requesting user.  *Application Note: This “distinct” path is merely invoked for the duration of its being needed (e.g., for re-authenticating the user); it need not be invoked for the duration of the interactive session.* |
|  |  | 6.3.1.2 | The OS shall permit users to initiate communication via a trusted path. |
|  |  | 6.3.1.3 | The OS shall require the use of a trusted path for all user operations involving authentication data. |
|  | **Management Requirements** | | |
|  |  | 6.3.2.x | None. |
|  | **Audit Requirements** | | |
|  |  | 6.3.3.x | None. |
|  | | | |
| **Revocation: Access Revocation**  The OS needs to assist an authorized administrator to cover from the situation where a user already has had access to the system, data, services, or resources, but that access needs to be revoked. | | | |
|  | **Functional Requirements** | | |
|  |  | 7.1.1.1 | The OS shall enforce the revocation of security relevant attributes associated with named objects when access checks are made. |
|  |  | 7.1.1.2 | The OS shall enforce the revocation of security relevant attributes associated with users at user session establishment. |
|  |  | 7.1.1.3 | The OS shall immediately terminate all subjects associated with deleted user accounts. |
|  | **Management Requirements** | | |
|  |  | 7.1.2.1 | The OS shall allow authorized administrators, object owners, and users with the DAC change attribute the ability to revoke security attributes associated with Discretionary Access Control policies on named objects. |
|  |  | 7.1.2.2 | The OS shall allow authorized administrators and subjects with the MIC change attribute the ability to revoke security attributes associated with Mandatory Integrity Control policies on named objects. |
|  |  | 7.1.2.3 | The OS shall allow authorized administrators the ability to revoke security attributes associated with users. |
|  |  | 7.1.2.4 | The OS shall allow authorized administrators the ability to delete user accounts. |
|  | **Audit Requirements** | | |
|  |  | 7.1.3.1 | The OS shall provide the ability to audit the revocation of security attribute. |
|  | | | |
| **Auditing: Audit Collection** | | | |
|  | **Functional Requirements** | | |
|  |  | 8.1.1.1 | The OS shall be able to generate audit records for all security-relevant events identified in this Commercial Grade OS Requirement Set and the other specific security relevant auditable events designed to be generated by the OS claiming compliance with this Commercial Grade OS Requirement Set.  *Application Note: The minimum set of security-relevant events for which audit must be possible is defined in the “Audit Functions” sections of each security functional requirement that the operating system satisfies.* |
|  |  | 8.1.1.2 | The OS shall be able to associate each auditable event with the identity or the user that caused the event.  *Application Note: For system startup, and in some cases system shutdown, no user identity association is required because the user is not under the control of the operating system. For failed logon attempts no user identity is required, except in cases that result in the locking of the user account.* |
|  |  | 8.1.1.3 | The OS shall be able to monitor and report the accumulation of specific sets of audit events known to indicate a potential security violation and immediately report the accumulated events when a threshold is exceeded.  The Commercial Grade OS Requirement Set requires the accumulation of the following events for reporting:   1. an administrator specified number of individual user authentication failures within an administrator specified time period; 2. an administrator specified number of Discretionary Access Control policy violation attempts by an individual user within an administrator specified time period; 3. an administrator specified number of Mandatory Integrity Control policy violation attempts by an individual user within an administrator specified time period; 4. any self-tests failure. |
|  |  | 8.1.1.4 | The OS shall record, within each audit record, the specific set of information items, as appropriate to the audit event.  The Commercial Grade OS Requirement Set requires the following set of information items to be recorded within an audit event record, as appropriate to the corresponding audit event:   1. date, time, and location of the event; 2. type of event; 3. event outcome (success or failure); 4. name of the object; 5. old and new values (except for authentication data and critical cryptographic security parameters) of changed security relevant data. |
|  | **Management Requirements** | | |
|  |  | 8.1.2.1 | The OS shall provide an authorized administrator with the capability to manage the threshold values specified in “8.1.1.3”. |
|  |  | 8.1.2.2 | The OS shall allow an authorized administrator to specify which events, from the set of auditable events, are to be audited. |
|  | **Audit Requirements** | | |
|  |  | 8.1.3.1 | The OS shall provide the ability to audit the modification of the threshold values specified in “8.1.1.3”. |
|  | | | |
| **Auditing: Audit Storage** | | | |
|  | **Functional Requirements** | | |
|  |  | 8.2.1.1 | The OS shall prevent modification of previously written audit records. |
|  |  | 8.2.1.2 | The OS shall provide the capability for authorized administrators to specify the specific actions to be taken upon audit storage exhaustion.  The Commercial Grade OS Requirement Set suggests the following set of actions for an administrator to select:   1. stop performing operations that are being audited; 2. overwrite oldest audit data; 3. automatically increase audit storage space; 4. automatically archive audit data; 5. disable auditing and continue to operate. |
|  | **Management Requirements** | | |
|  |  | 8.2.2.1 | The OS shall provide an authorized administrator with the capability to specify actions to be taken upon audit storage exhaustion. |
|  |  | 8.2.2.2 | The OS shall provide an authorized administrator with the capability to sort, select and review collected audit records based on identity and any audit information items specified in “8.1.1.4”. |
|  |  | 8.2.2.3 | The OS shall provide an authorized administrator with the capability to archive audit data. |
|  | **Audit Requirements** | | |
|  |  | 8.2.3.1 | The OS shall provide the ability to audit the deleting or archiving of audit data. |

# Summary of Windows’ Compliance with the Commercial Grade Operating System Security Functional Requirement Set

While this paper aims to present the Windows Operating System (Windows OS) specific technical justifications for the compliance with individual requirements in the subsequent sections, we also provide a summary of their compliances in the following table.

|  |  |  |  |
| --- | --- | --- | --- |
| **Req. No.** | **Individual requirement brief description** | **Windows OS Compliance:** | **Note:** |
| 1.1.1.1 | The OS shall protect security-relevant data from unauthorized access. | Addressed  - in Page 58. | The Windows OS addresses this requirement based on the concept of “defense in depth” for achieving multiple levels of protections. |
| 1.1.1.2 | The OS shall ensure that all security policies are enforced before each security function is allowed to proceed. | Met  - in Page 67. |  |
| 1.1.1.3 | The OS shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects. | Met.  - in Page 83. |  |
| 1.1.1.4 | The OS shall enforce separation between the security domains of subjects. | Met.  - in Page 88. |  |
| 1.1.1.5 | The OS shall make effective use of hardware provided security features. | Addressed.  - in Page 90. |  |
| 1.1.2.x | None. |  |  |
| 1.1.3.x | None. |  |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 1.2.1.1 | The OS shall ensure that security-relevant data is consistent between parts of the OS by providing a mechanism to bring inconsistent data into a consistent state in a timely manner. | Met.  - in Page 95. | The Windows OS uses the capabilities of its Directory Replication Service (DRS) and File System Replication (DFSR) only to meet this requirement. |
| 1.2.1.2 | The OS shall protect security-relevant data from disclosure while being transmitted to a remote part of the OS through the use of OS provided cryptographic services. | Met.  - in Page 110. | A pure Windows Vista and Windows Server 2008 environment is necessary. |
| 1.2.1.3 | The OS shall detect modification and insertion of security-relevant received from a remote part of the OS through the use of OS provided cryptographic services. | Met.  - in Page 114. | A pure Windows Vista and Windows Server 2008 environment is necessary. |
| 1.2.1.4 | Upon detection of modification and insertion of security-relevant received from a remote part of the OS, the OS shall reject the data. | Met.  - in Page 114. | A pure Windows Vista and Windows Server 2008 environment is necessary. |
| 1.2.1.5 | The OS shall provide a means for secure remote administration using OS provided cryptographic services. | Met.  - in Page 115. | A pure Windows Vista and Windows Server 2008 environment is necessary. This requirement and “1.2.2.1” are addressed together. |
| 1.2.1.6 | The OS shall authenticate remote parts of the OS using OS provided cryptographic services. | Met.  - in Page 115. | A pure Windows Vista and Windows Server 2008 environment is necessary. |
| 1.2.2.1 | The OS shall provide the ability for an authorized administrator to remotely manage the OS. | Met.  - in Page 117. | A pure Windows Vista and Windows Server 2008 environment is necessary. This requirement and “1.2.1.5” are addressed together. |
| 1.2.3.1 | The OS shall provide the ability to audit when modification or insertion of security-relevant data received from a remote part of the OS has been detected. | Met.  - in Page 126. | A pure Windows Vista and Windows Server 2008 environment is necessary. |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 2.1.1.1 | The OS shall enforce a Discretionary Access Control policy on all subjects, all named objects and all access operations among them. | Addressed.  - in Page 127. |  |
| 2.1.1.2 | The OS shall associate Discretionary Access Control security attributes with each subject and named object. | Met.  - in Page 135. |  |
| 2.1.1.3 | The OS shall enforce the Discretionary Access Control policy on named objects based on the specific types of the subject and object security attributes. | Met.  - in Page 139. |  |
| 2.1.1.4 | The OS shall provide restrictive default values for object security attributes that are used to enforce the Discretionary Access Control policy. | Met.  - in Page 139. |  |
| 2.1.1.5 | The OS shall ensure that only valid values are accepted for an object’s Discretionary Access Control security attributes. | Met.  - in Page 140. |  |
| 2.1.1.6 | The Discretionary Access Control policy mechanism shall provide that named objects are protected from unauthorized access according to a specific set of ordered rules. | Addressed.  - in Page 140. | The Windows OS rules include the “object security inheritance” concept also. |
| 2.1.2.1 | The OS shall allow only authorized administrators, object owners, and users with the DAC change authorization the ability to change the access permission associated with a named object. | Met.  - in Page 148. |  |
| 2.1.2.2 | The OS shall allow only authorized administrators the ability to change object ownership. | Met.  - in Page 148. |  |
| 2.1.2.3 | The OS shall provide only authorized administrators the ability to specify alternative initial values for default Discretionary Access Control object security attributes. | Met.  - in Page 149. |  |
| 2.1.3.1 | The OS shall provide the ability to audit all Discretionary Access Control policy decisions. | Addressed.  - in Page 150. |  |
| 2.1.3.2 | The OS shall provide the ability to audit all changes to an object’s Discretionary Access Control security attributes. | Partially addressed.  - in Page 153. | Some user mode resource managers do not necessary generate a security descriptor change audit record for a non standard user owned named object, after the change is permitted to proceed. However, the access check for the permission to make the security descriptor change is audited. |
| 2.1.3.3 | The OS shall provide the ability to audit all modifications to default Discretionary Access Control object security attribute values. | Addressed with a minor exception.  - in Page 155. | Using the TOKEN\_ADJUST\_DEFAULT access to change the default DACL in the primary or impersonation access token of an object's creator subject by an administrator is not audited for indicating the actual change. However, the default DACL of an access token is not typically changed. |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 2.2.1.1 | The OS shall enforce a Mandatory Integrity Control policy on untrusted subjects, named objects, and network interfaces and all operations that cause information to flow among them. | Addressed with an exception.  - in Page 156. | In Windows OS, network interfaces are not covered by the Mandatory Integrity Control (MIC) policy. We believe that the intent of the network interface MIC policy enforcement could be overcome through the capabilities of the Windows OS Network List Manager (NLM) and window terminal session separation. |
| 2.2.1.2 | The OS shall associate an integrity label with each subject, named object, and network interface that accurately represents its integrity level. | Met.  - in Page 157. |  |
| 2.2.1.3 | The OS shall enforce the Mandatory Integrity Control policy based on the specific types of the subject and object integrity attributes. | Met.  - in Page 161. |  |
| 2.2.1.4 | The OS shall enforce specific relationships for any two valid integrity labels. | Met.  - in Page 162. |  |
| 2.2.1.5 | The OS shall permit an information flow among subjects and objects based on a specific set of rules. | Addressed.  - in Page 163. | A workaround for the “no read down” [i.e. Action b)] rule for files is provided through the “no write up” [i.e. Action a)] rule and the capabilities of the Windows OS Administrative Privileged Application Launching Service. |
| 2.2.1.6 | The OS shall enforce that all definition of integrity labels are unique. | N/A.  - in Page 166. | The Windows OS supports only a hierarchical integrity policy scheme. |
| 2.2.2.1 | The OS shall provide the ability to set the system-wide definition of integrity labels to authorized administrators. | N/A.  - in Page 167. | The hierarchical integrity policy scheme supported by the Windows OS is fixed. |
| 2.2.2.2 | The OS shall provide authorized administrators the ability to change integrity labels of subjects and objects. | Met.  - in Page 167. |  |
| 2.2.3.1 | The OS shall provide the ability to audit all Mandatory Integrity Control policy decisions. | Addressed.  - in Page 168. |  |
| 2.2.3.2 | The OS shall provide the ability to audit the setting and changing of system-wide integrity label definitions. | N/A.  - in Page 168. | The hierarchical integrity policy scheme supported by the Windows OS is fixed. |
| 2.2.3.3 | The OS shall provide the ability to audit the changing of subject and object integrity labels. | Addressed with a minor exception.  - in Page 168. | The use of the SeTcbPrivilege to increase the integrity level of an existing (Windows OS) access token of a subject is not audited. By default, the SeTcbPrivilege is assigned only to the local system. |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 3.1.1.1 | The OS shall require each user to be uniquely identified and successfully authenticated by means of a password before allowing any actions on behalf of that user. | Addressed with certain minor exceptions.  - in Page 169. | The minor exceptions are necessary for supporting interoperability and general usability, in the case where an unauthenticated user needs to look up “public” information within the network.  In addition to password logon, the Windows OS also supports smart card logon. |
| 3.1.1.2 | The OS shall be able to support passwords up to 32 characters in length, consisting of any combination of upper and lower case letters, numbers, and punctuations. | Met.  - in Page 174. |  |
| 3.1.1.3 | The OS authentication mechanism shall provide a specific set of capabilities. | Met.  - in Page 175. |  |
| 3.1.1.4 | The OS shall provide only obscured feedback to the user while the authentication is in progress. | Met.  - in Page 176. |  |
| 3.1.1.5 | The OS shall detect when an authorized administrator specified positive integer of consecutive unsuccessful authentication attempts occur related to any authorized user authentication process. | Met.  - in Page 176. |  |
| 3.1.1.6 | The OS shall perform a specific set of actions when the defined number of consecutive unsuccessful authentication attempts specified in “3.1.1.5” has been detected. | Addressed.  - in Page 178. |  |
| 3.1.1.7 | The OS shall maintain a specific list of security attributes belonging to individual users. | Met.  - in Page 180. |  |
| 3.1.1.8 | The OS shall associate a specific list of user security attributes with subjects acting on behalf of that user. | Met.  - in Page 189. |  |
| 3.1.1.9 | The OS shall protect authentication data from disclosure through the use of operating system provided cryptographic services. | Addressed.  - in Page 196. |  |
| 3.1.1.10 | The OS shall provide an administrator role that is separate from untrusted users. | Met.  - in Page 209. |  |
| 3.1.1.11 | The OS shall provide the authorized administrator the ability to set user attributes. | Met.  - in Page 210. |  |
| 3.1.1.12 | The OS shall re-authenticate the user when changing authentication data. | Met.  - in Page 213. |  |
| 3.1.1.13 | The OS shall provide a mechanism to verify that each user password when set meets authorized administrator configurable password characteristics. | Met.  - in Page 213. |  |
| 3.1.1.14 | The OS shall automatically disable a user account when the current time has passed the authorized administrator specified time interval in which the user account has not been logged on successfully since the user account’s last successful logon time. | Addressed.  - in Page 214. | We suggest that the Windows OS “maximum password age” policy is as capable as the enforcement due to the elapse of the time interval in which the user account has not been logged on successfully since the user account’s last successful logon time. |
| 3.1.2.1 | The OS shall allow only authorized administrators the ability to create and manage user accounts. | Met.  - in Page 216. |  |
| 3.1.2.2 | The OS shall provide authorized administrators the ability to specify a time interval in which the user account has not been logged on successfully so that a user account is automatically disabled after the time interval has elapsed. | Addressed.  - in Page 216. | We suggest that the Windows OS “maximum password age” policy is as capable as the enforcement due to the elapse of the time interval in which the user account has not been logged on successfully since the user account’s last successful logon time. |
| 3.1.2.3 | The OS shall allow only authorized administrators the ability to initially set and modify user security attributes (other than authentication data). | Met.  - in Page 217. |  |
| 3.1.2.4 | The OS shall allow only authorized administrators the ability to initialize user authentication data. | Met.  - in Page 218. |  |
| 3.1.2.5 | The OS shall restrict the ability to modify authentication data to authorized administrators and users authorized to modify their own authentication data. | Met.  - in Page 218. |  |
| 3.1.2.6 | The OS shall allow only authorized administrators the ability to initialize and modify authentication mechanism attributes. | Met.  - in Page 219. |  |
| 3.1.2.7 | The OS shall provide authorized administrators the ability to specify and configure mandatory password composition. | Met.  - in Page 223. |  |
| 3.1.3.1 | The OS shall provide the ability to audit the creation and management of user accounts. | Met.  - in Page 224. |  |
| 3.1.3.2 | The OS shall provide the ability to audit the initialization and modification of user security attributes. | Met.  - in Page 226. |  |
| 3.1.3.3 | The OS shall provide the ability to audit the initialization and modification of authentication mechanism attributes. | Met.  - in Page 229. |  |
| 3.1.3.4 | The OS shall provide the ability to audit when a user’s consecutive unsuccessful authentication attempts meets or exceeds the administrator-configured positive integer. | Met.  - in Page 231. |  |
| 3.1.3.5 | The OS shall provide the ability to audit all user attempts to identify and authenticate to the system. | Met.  - in Page 231. |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 3.2.1.1 | Before establishing an interactive session, the OS shall display an authorized administrator specified advisory notice. | Met.  - in Page 238. |  |
| 3.2.1.2 | The OS shall allow user-initiated locking of an interactive session by performing a specific set of actions. | Met.  - in Page 241. |  |
| 3.2.1.3 | The OS shall require a specific set of actions from users and administrators. | Addressed.  - in Page 271. | Action b) of this requirement seems to be a side-effect consequence of a “single user only” operating system. This action is handled differently on the Windows OS. With the Windows OS “Fast User Switching” functionality, a window terminal session does not need to be terminated in order to let another user logon to a different window terminal session of the same machine. |
| 3.2.1.4 | The OS shall enforce an authorized administrator specified maximum number of concurrent interactive sessions per user. | Addressed.  - in Page 304. | A workaround is necessary due to a potential denial of service (DOS) attack induced by this requirement in an environment, where a high reliability level of the underlying (wired or wireless) network transport medium is not guaranteed. |
| 3.2.1.5 | The OS shall lock an interactive session after an authorized administrator specified time interval of user inactivity by performing a specific set of actions. | Met.  - in Page 305. |  |
| 3.2.1.6 | The OS shall provide the ability to deny interactive session establishment based on time and day. | Met.  - in Page 321. |  |
| 3.2.1.7 | Upon successful interactive session establishment, the OS shall display to the authorized user a specific set of user access history information elements. | Addressed with a minor exception.  - in Page 323. | The minor exception is that the last successful logon location is not available for displaying to the logging on user. |
| 3.2.1.8 | The OS shall not erase the user access history information elements from the authorized user interface without giving the authorized user the opportunity to review the information elements. | Met.  - in Page 325. |  |
| 3.2.2.1 | The OS shall provide authorized administrators with the ability to specify an advisory notice. | Met.  - in Page 327. |  |
| 3.2.2.2 | The OS shall provide authorized administrators with the ability to terminate an interactive session. | Met.  - in Page 327. |  |
| 3.2.2.3 | The OS shall provide authorized administrators with the ability to specify the number of concurrent interactive sessions allowed per user. | Addressed.  - in Page 336. | With the consideration of the workaround for addressing “3.2.1.4”. |
| 3.2.2.4 | The OS shall provide authorized administrators with the ability to deny interactive session establishment based on system parameters specified in “3.2.1.6”. | Met.  - in Page 337. |  |
| 3.2.2.5 | The OS shall provide authorized administrators with the ability to specify a time interval of session inactivity after which an (inactive) interactive session is locked. | Met.  - in Page 338. |  |
| 3.2.3.1 | The OS shall provide the ability to audit all user attempts to re-authenticate to the system (for the interactive session unlocking purpose). | Met.  - in Page 339. |  |
| 3.2.3.2 | The OS shall provide the ability to audit all administrator attempts to terminate a user’s locked session. | Addressed.  - in Page 339. |  |
| 3.2.3.3 | The OS shall provide the ability to audit any attempt to exceed the maximum number of concurrent interactive sessions by a user. | Addressed.  - in Page 341. | With the consideration of the workaround for addressing “3.2.1.4”. |
| 3.2.3.4 | The OS shall provide the ability to audit all user attempts that violate the restrictions specified in “3.2.1.6”. | Met.  - in Page 342. |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 4.1.1.1 | The OS shall provide a specific list of cryptographic services to applications. | Addressed.  - in Page 344. |  |
| 4.1.2.x | None. |  |  |
| 4.1.3.x | None. |  |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 4.2.1.1 | The OS shall provide FIPS-approved (i.e. implying also ISO/IEC compliant) cryptographic algorithms and modes of operations, implemented in a crypto module that is FIPS 140-2 Security Level 2 validated (i.e. implying also ISO/IEC 19790 compliant). | Addressed.  - in Page 347. |  |
| 4.2.1.2 | A specific list of algorithms shall be provided for the corresponding cryptographic services. | Met.  - in Page 352. |  |
| 4.2.2.1 | The OS shall provide the authorized administrators the ability to select (when more than one algorithm is available) the appropriate algorithm for encryption/decryption, digital signature, hashing, and key agreement. | Addressed.  - in Page 358. |  |
| 4.2.3.x | None. |  |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 4.3.1.1 | The OS crypto module shall provide Random Number Generation (RNG) services in accordance with a FIPS-Approved RNG listed in [FIPS 140-2 Annex C] composed of a specific method. | Addressed, with certain minor exceptions.  - in Page 360. | Currently, neither the Windows OS Cryptographic Primitives Library (bcrypt.dll) nor the Windows OS kernel security device driver (ksecdd.sys) has yet received the corresponding DRNG algorithm validation certificates from NIST CAVP for their implementations of the NIST SP 800-90 EC\_DRGB and AES CTR\_DRBG algorithms. |
| 4.3.2.x | None. |  |  |
| 4.3.3.x | None. |  |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 4.4.1.1 | The OS crypto module shall generate symmetric cryptographic keys using a random number generator as specified in “4.3.1.1” | Addressed.  - in Page 365. |  |
| 4.4.1.2 | The OS crypto module shall generate asymmetric cryptographic keys using a domain parameter generator and a random number generator as specified in “4.3.1.1”. | Addressed, with a minor exception.  - in Page 365. | NIST CAVP accepts Microsoft’s vendor-affirmed prerequisite NIST SP 800-90 RNG implementation that underlies the asymmetric cryptographic key generations as the Windows OS Cryptographic Primitives Library (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from NIST CAVP for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation. |
| 4.4.1.3 | The OS crypto module shall generate cryptographic keys such that generated key strength shall be equivalent to, or greater than, a symmetric key strength of 128 bits using conservative estimates as specified in [NIST SP 800-57 Section 5.6.1]. | Addressed, with a minor exception.  - in Page 368. | NIST CAVP accepts Microsoft’s vendor-affirmed prerequisite NIST SP 800-90 RNG implementation that underlies the asymmetric cryptographic key generations as the Windows OS Cryptographic Primitives Library (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from NIST CAVP for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation. |
| 4.4.1.4 | The OS crypto module shall perform key entry and output in accordance with the Level 1 Key Entry and Output requirements in [FIPS 140-2], “Security Requirements for Cryptographic Modules”. | Met.  - in Page 372. |  |
| 4.4.1.5 | The OS crypto module shall provide the ability to protect secret keys, private keys, and critical cryptographic security parameters from unauthorized disclosure, modification, and substitution. | Addressed.  - in Page 373. | Long lived public / private key pairs and calculated secret agreement values are subject to the protection of this requirement in the Windows OS. |
| 4.4.1.6 | The OS crypto module shall provide the ability to protect public keys from unauthorized modification, and substitution. | Met.  - in Page 389. |  |
| 4.4.1.7 | The OS crypto module shall provide the ability to correctly associate stored cryptographic keys (secret, private, or public) with the entity (e.g. person, group, or subject) to which the key is assigned. | Met.  - in Page 392. |  |
| 4.4.1.8 | The OS crypto module shall destroy cryptographic keys in accordance with a specific cryptographic key zeroization method. | Addressed.  - in Page 393. |  |
| 4.4.2.x | None. |  |  |
| 4.4.3.1 | The OS shall provide the ability to audit attempts to violate the restrictions specified in “4.4.1.5”. | Met.  - in Page 396. | Long lived public / private key pairs and calculated secret agreement values are subject to the protection of the “4.4.1.5”requirement in the Windows OS. |
| 4.4.3.2 | The OS shall provide the ability to audit attempts to violate the restrictions specified in “4.4.1.6”. | Met.  - in Page 398. |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 4.5.1.1 | The OS crypto module shall run a suite of self-tests to demonstrate the correct operation of the cryptographic functions in accordance with [FIPS 140-2 Section 4.9] during specific occasions. | Addressed, with certain minor exceptions.  - in Page 400. | After the Windows OS Cryptographic Primitives Library (bcrypt.dll) and the Windows OS kernel security device driver (ksecdd.sys) have passed the power-on (start up) self-tests once and have been successfully loaded, they do not re-perform the power-on (start up) tests again either at the request of an authorized administrator (on demand) or periodically (at least once a day). |
| 4.5.1.2 | If any of the cryptographic self-tests fail, the OS shall react as required by [FIPS 140-2 Section 4.9] for failing a self-test. | Met.  - in Page 406. |  |
| 4.5.1.3 | The OS crypto module shall run a set of specific additional RNG tests. | Addressed, with a minor exception.  - in Page 408. | After the Windows OS Cryptographic Primitives Library (bcrypt.dll) and the Windows OS kernel security device driver (ksecdd.sys) have passed the power-on (start up) known answer RNG tests of [NIST SP 800-90] once and have been successfully loaded, they do not re-perform the power-on (start up) tests again at the request of an authorized administrator (on demand). |
| 4.5.2.1 | The OS shall provide the ability for an authorized administrator to run a suite of crypto module self-tests and RNG tests. | Addressed, with a minor exception.  - in Page 412. | After the Windows OS Cryptographic Primitives Library (bcrypt.dll) and the Windows OS kernel security device driver (ksecdd.sys) have passed the power-on (start up) self-tests and known answer RNG tests of [NIST SP 800-90] once and have been successfully loaded, they do not re-perform the power-on (start up) tests again at the request of an authorized administrator (on demand). |
| 4.5.3.1 | The OS shall audit failure of any crypto module self-tests and RNG tests. | Addressed.  - in Page 413. |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 5.1.1.1 | The OS shall ensure that any previous information content of a resource is made unavailable upon either the allocation of the resource to or the de-allocation of the resource from all non cryptographic objects. | Addressed.  - in Page 415. |  |
| 5.1.1.2 | Objects associated with cryptographic keys and critical cryptographic security parameters shall be destroyed as described in “4.4.1.8”. | Addressed, with a minor exception.  - in Page 418. | Only the Windows OS Cryptographic Primitives Library (bcrypt.dll) does the read-zero-verify check after making the overwriting with zeros. The other components only make the overwriting with zeros. |
| 5.1.2.x | None. |  |  |
| 5.1.3.x | None. |  |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 5.2.1.1 | The OS shall enforce maximum quotas on the portion of shared persistent storage that individual authorized users can use. | Met.  - in Page 423. |  |
| 5.2.2.1 | The OS shall allow only authorized administrators the ability to set maximum quotas on shared persistent storage. | Met.  - in Page 427. |  |
| 5.2.3.1 | The OS shall provide the ability to audit any request by a user that attempts to exceed the authorized administrator defined shared persistent storage quota. | Met.  - in Page 428. |  |
| 5.2.3.2 | The OS shall provide the ability to audit the setting of maximum quotas on shared persistent storage. | Not met.  - in Page 429. | We recommend (as a workaround for this “5.2.3.2” requirement) that an organization monitors the setting of the user quota information entry for a specific user account and the setting of the default quota information entry of specific targeted NTFS non read-only volumes for changes. |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 5.3.1.1 | None. |  |  |
| 5.3.1.2 | None. |  |  |
| 5.3.1.3 | The OS shall verify during initial startup the integrity of executable code that implements access control and cryptographic functionality through the use of the OS system provided cryptographic services. | Addressed, with a minor exception.  - in Page 430. | The user mode authorization framework library (authz.dll) is not checked for its integrity cryptographically as an image file before it is loaded into memory. |
| 5.3.2.x | None. |  |  |
| 5.3.3.3 | The OS shall provide the ability to audit the failure of the (executable code) integrity verification. | Met.  - in Page 442. |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 6.1.1.1 | The OS shall ensure that security attributes on named objects, when exported to removable media, are associated with the object. | Met.  - in Page 444. |  |
| 6.1.2.x | None. |  |  |
| 6.1.3.1 | The OS shall provide the ability to audit the exportation of named objects to removable media. | Addressed.  - in Page 446. |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 6.2.1.1 | The OS shall enforce a specific set of rules when importing user data. | Met.  - in Page 448. |  |
| 6.2.2.1 | The OS shall provide authorized administrators with the ability to select the technique for handling imported data with attributes that cannot be validated. | N/A.  - in Page 454. | The corresponding behaviors in the Windows OS are fixed. |
| 6.2.3.1 | The OS shall provide the ability to audit the failure of security attribute validation. | Addressed.  - in Page 455. | In the case of an attempted import of the clear-text file data belonging to an EFS-protected named file, this auditing is deemed as unnecessary as the outcome is not sufficiently different from any other kind of user data unavailability situations from the end-user experience perspective. |
| 6.2.3.2 | The OS shall provide the ability to audit the administrator selected configuration for handling imported data with attributes that cannot be validated. | N/A.  - in Page 455. | The corresponding behaviors in the Windows OS are fixed. |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 6.3.1.1 | The OS shall provide a communication path between itself and users that is logically distinct from other communication paths and provides assured identification of the OS to the requesting user. | Met.  - in Page 456. |  |
| 6.3.1.2 | The OS shall permit users to initiate communication via a trusted path. | Met.  - in Page 462. |  |
| 6.3.1.3 | The OS shall require the use of a trusted path for all user operations involving authentication data. | Met.  - in Page 463. |  |
| 6.3.2.x | None. |  |  |
| 6.3.3.x | None. |  |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 7.1.1.1 | The OS shall enforce the revocation of security relevant attributes associated with named objects when access checks are made. | Met.  - in Page 467. |  |
| 7.1.1.2 | The OS shall enforce the revocation of security relevant attributes associated with users at user session establishment. | Met.  - in Page 467. |  |
| 7.1.1.3 | The OS shall immediately terminate all subjects associated with deleted user accounts. | Addressed, with a minor exception.  - in Page 472. | If the user account has already been logged on before its deletion occurs, there is no guarantee that the Windows OS machine, where this user account was logged on, is still connected. We recommend (as a workaround for this “7.1.1.3” requirement) that the authorized administrator should separately terminate the subjects (associated with the deleted user account) which are still active in the logged on Windows OS machines, when the connections to the Windows OS machines become available. |
| 7.1.2.1 | The OS shall allow authorized administrators, object owners, and users with the DAC change attribute the ability to revoke security attributes associated with Discretionary Access Control policies on named objects. | Met.  - in Page 475. |  |
| 7.1.2.2 | The OS shall allow authorized administrators and subjects with the MIC change attribute the ability to revoke security attributes associated with Mandatory Integrity Control policies on named objects. | Met.  - in Page 475. |  |
| 7.1.2.3 | The OS shall allow authorized administrators the ability to revoke security attributes associated with users. | Met.  - in Page 476. |  |
| 7.1.2.4 | The OS shall allow authorized administrators the ability to delete user accounts. | Met.  - in Page 477. |  |
| 7.1.3.1 | The OS shall provide the ability to audit the revocation of security attribute. | Met.  - in Page 479. |  |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 8.1.1.1 | The OS shall be able to generate audit records for all security-relevant events identified in this Commercial Grade OS Requirement Set and the other specific security relevant auditable events designed to be generated by the OS claiming compliance with this Commercial Grade OS Requirement Set. | Addressed.  - in Page 481. |  |
| 8.1.1.2 | The OS shall be able to associate each auditable event with the identity or the user that caused the event. | Addressed.  - in Page 481. |  |
| 8.1.1.3 | The OS shall be able to monitor and report the accumulation of specific sets of audit events known to indicate a potential security violation and immediately report the accumulated events when a threshold is exceeded. | Addressed.  - in Page 481. | The requirement is addressed by the Windows OS with the capabilities provided by [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx). |
| 8.1.1.4 | The OS shall record, within each audit record, the specific set of information items, as appropriate to the audit event. | Addressed.  - in Page 483. |  |
| 8.1.2.1 | The OS shall provide an authorized administrator with the capability to manage the threshold values specified in “8.1.1.3”. | Addressed.  - in Page 485. | The requirement is addressed by the Windows OS with the capabilities provided by [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx). |
| 8.1.2.2 | The OS shall allow an authorized administrator to specify which events, from the set of auditable events, are to be audited. | Met.  - in Page 486. |  |
| 8.1.3.1 | The OS shall provide the ability to audit the modification of the threshold values specified in “8.1.1.3”. | Addressed.  - in Page 495. | The requirement is addressed by the Windows OS with the capabilities provided by [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx). |
| **\*\* \*\*\*\*\*\*\*\* \*\*** | | | |
| 8.2.1.1 | The OS shall prevent modification of previously written audit records. | Met.  - in Page 496. |  |
| 8.2.1.2 | The OS shall provide the capability for authorized administrators to specify the specific actions to be taken upon audit storage exhaustion. | Met.  - in Page 504. |  |
| 8.2.2.1 | The OS shall provide an authorized administrator with the capability to specify actions to be taken upon audit storage exhaustion. | Met.  - in Page 507. |  |
| 8.2.2.2 | The OS shall provide an authorized administrator with the capability to sort, select and review collected audit records based on identity and any audit information items specified in “8.1.1.4”. | Met.  - in Page 507. |  |
| 8.2.2.3 | The OS shall provide an authorized administrator with the capability to archive audit data. | Met.  - in Page 508. |  |
| 8.2.3.1 | The OS shall provide the ability to audit the deleting or archiving of audit data. | Met.  - in Page 510. |  |

Contents

[Introduction 3](#_Toc225063908)

[A Commercial Grade Operating System Security Functional Requirement Set 5](#_Toc225063909)

[Summary of Windows’ Compliance with the Commercial Grade Operating System Security Functional Requirement Set 25](#_Toc225063910)

[Meeting the “Security Architecture Separation, Isolation, and Least Privilege Functional Requirements” 60](#_Toc225063911)

[Addressing 1.1.1.1 “The OS shall protect security-relevant data from unauthorized access” 60](#_Toc225063912)

[Full volume encryption 61](#_Toc225063913)

[User authentication 61](#_Toc225063914)

[User mode kernel mode context switching 62](#_Toc225063915)

[Process memory virtualization 64](#_Toc225063916)

[Window terminal session isolation 64](#_Toc225063917)

[Secure desktop separation 66](#_Toc225063918)

[Isolation of user-entered credential through encryption 66](#_Toc225063919)

[User interface privilege isolation within the same desktop 67](#_Toc225063920)

[Persistent data access mediation 68](#_Toc225063921)

[Encrypting file system (EFS) 68](#_Toc225063922)

[Long lived cryptographic key isolation service 69](#_Toc225063923)

[Addressing 1.1.1.2 “The OS shall ensure that all security policies are enforced before each security function is allowed to proceed” 69](#_Toc225063924)

[Full volume encryption 69](#_Toc225063925)

[User authentication 74](#_Toc225063926)

[User mode kernel mode context switching 79](#_Toc225063927)

[Process memory virtualization 80](#_Toc225063928)

[Window terminal session isolation 80](#_Toc225063929)

[Secure desktop separation 81](#_Toc225063930)

[Isolation of user-entered credential through encryption 82](#_Toc225063931)

[User interface privilege isolation within the same desktop 82](#_Toc225063932)

[Persistent data access mediation 83](#_Toc225063933)

[Encrypting file system (EFS) 84](#_Toc225063934)

[Long lived cryptographic key isolation service 85](#_Toc225063935)

[Addressing 1.1.1.3 “The OS shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects” 85](#_Toc225063936)

[Full volume encryption 86](#_Toc225063937)

[User authentication 86](#_Toc225063938)

[User mode kernel mode context switching 86](#_Toc225063939)

[Process memory virtualization 87](#_Toc225063940)

[Window terminal session isolation 87](#_Toc225063941)

[Secure desktop separation 87](#_Toc225063942)

[Isolation of user-entered credential through encryption 88](#_Toc225063943)

[User interface privilege isolation within the same desktop 88](#_Toc225063944)

[Persistent data access mediation 88](#_Toc225063945)

[Encrypting file system (EFS) 89](#_Toc225063946)

[Long lived cryptographic key isolation service 90](#_Toc225063947)

[Addressing 1.1.1.4 “The OS shall enforce separation between the security domains of subjects” 90](#_Toc225063948)

[Full volume encryption 90](#_Toc225063949)

[User authentication 90](#_Toc225063950)

[User mode kernel mode context switching 90](#_Toc225063951)

[Process memory virtualization 90](#_Toc225063952)

[Window terminal session isolation 91](#_Toc225063953)

[Secure desktop separation 91](#_Toc225063954)

[Isolation of user-entered credential through encryption 91](#_Toc225063955)

[User interface privilege isolation within the same desktop 91](#_Toc225063956)

[Persistent data access mediation 92](#_Toc225063957)

[Encrypting file system (EFS) 92](#_Toc225063958)

[Long lived cryptographic key isolation service 92](#_Toc225063959)

[Addressing 1.1.1.5 “The OS shall make effective use of hardware provided security features” 92](#_Toc225063960)

[Full volume encryption 92](#_Toc225063961)

[User authentication 92](#_Toc225063962)

[User mode kernel mode context switching 92](#_Toc225063963)

[Process memory virtualization 93](#_Toc225063964)

[Window terminal session isolation 93](#_Toc225063965)

[Secure desktop separation 93](#_Toc225063966)

[Isolation of user-entered credential through encryption 93](#_Toc225063967)

[User interface privilege isolation within the same desktop 93](#_Toc225063968)

[Persistent data access mediation 93](#_Toc225063969)

[Encrypting file system (EFS) 93](#_Toc225063970)

[Long lived cryptographic key isolation service 93](#_Toc225063971)

[Meeting the “Security Architecture Separation, Isolation, and Least Privilege Management Requirements” 95](#_Toc225063972)

[Meeting the “Security Architecture Separation, Isolation, and Least Privilege Audit Requirements” 96](#_Toc225063973)

[Meeting the “Security Architecture Distributed Architectures Functional Requirements” 97](#_Toc225063974)

[Addressing 1.2.1.1 “The OS shall ensure that security-relevant data is consistent between parts of the OS by providing a mechanism to bring inconsistent data into a consistent state in a timely manner” 97](#_Toc225063975)

[Windows OS Active Directory (AD) 97](#_Toc225063976)

[Active Directory Replication 97](#_Toc225063977)

[File Replication 98](#_Toc225063978)

[Multi-master loose consistency with convergence 98](#_Toc225063979)

[Addressing 1.2.1.2 “The OS shall protect security-relevant data from disclosure while being transmitted to a remote part of the OS through the use of OS provided cryptographic services” 112](#_Toc225063980)

[RPC interface for DRS 112](#_Toc225063981)

[SMTP mail interface for DRS 113](#_Toc225063982)

[RPC interface for DFSR 113](#_Toc225063983)

[Crypto Systems of the Kerberos security provider 114](#_Toc225063984)

[Cryptography used in “UPDATE REPLICA” mails 115](#_Toc225063985)

[Addressing 1.2.1.3 “The OS shall detect modification and insertion of security-relevant received from a remote part of the OS through the use of OS provided cryptographic services” 116](#_Toc225063986)

[Addressing 1.2.1.4 “Upon detection of modification and insertion of security-relevant received from a remote part of the OS, the OS shall reject the data” 116](#_Toc225063987)

[Addressing 1.2.1.5 “The OS shall provide a means for secure remote administration using OS provided cryptographic services” 117](#_Toc225063988)

[Addressing 1.2.1.6 “The OS shall authenticate remote parts of the OS using OS provided cryptographic services” 117](#_Toc225063989)

[Meeting the “Security Architecture Distributed Architectures Management Requirements” 119](#_Toc225063990)

[Addressing 1.2.2.1 “The OS shall provide the ability for an authorized administrator to remotely manage the OS” 119](#_Toc225063991)

[Windows OS group policy 119](#_Toc225063992)

[Group policy processing at a local computer 124](#_Toc225063993)

[Transportation security of the LDAP connection to AD 126](#_Toc225063994)

[Transportation security of the SMB V2 connection to the replicated SysVol directory of an AD DC 126](#_Toc225063995)

[Meeting the “Security Architecture Distributed Architectures Audit Requirements” 128](#_Toc225063996)

[Addressing 1.2.3.1 “The OS shall provide the ability to audit when modification or insertion of security-relevant data received from a remote part of the OS has been detected” 128](#_Toc225063997)

[Meeting the “Access Control Polices Discretionary Access Control Policy Functional Requirements” 129](#_Toc225063998)

[Addressing 2.1.1.1 “The OS shall enforce a Discretionary Access Control policy on all subjects, all named objects and all access operations among them” 129](#_Toc225063999)

[Addressing 2.1.1.2 “The OS shall associate Discretionary Access Control security attributes with each subject and named object” 137](#_Toc225064000)

[Assignment of DACL to named objects which are not Active Directory objects with attributes 138](#_Toc225064001)

[Assignment of DACL to Active Directory objects with attributes 138](#_Toc225064002)

[Content of an ACE 139](#_Toc225064003)

[Identification of the owner of an object 140](#_Toc225064004)

[Addressing 2.1.1.3 “The OS shall enforce the Discretionary Access Control policy on named objects based on the specific types of the subject and object security attributes” 141](#_Toc225064005)

[Addressing 2.1.1.4 “The OS shall provide restrictive default values for object security attributes that are used to enforce the Discretionary Access Control policy” 141](#_Toc225064006)

[Addressing 2.1.1.5 “The OS shall ensure that only valid values are accepted for an object’s Discretionary Access Control security attributes” 142](#_Toc225064007)

[Addressing 2.1.1.6 “the Discretionary Access Control policy mechanism shall provide that named objects are protected from unauthorized access according to a specific set of ordered rules” 142](#_Toc225064008)

[Extended DAC policy ordered rules for object security inheritance 142](#_Toc225064009)

[Implementation locations of the Windows OS access determination algorithm 143](#_Toc225064010)

[Windows OS access determination algorithm 144](#_Toc225064011)

[Windows OS placement of inherited ACEs in the DACL of an object 145](#_Toc225064012)

[Ordering of ACEs in an object’s DACL 145](#_Toc225064013)

[Windows OS resource managers following the rules of the Windows OS DACL preferred ACE order 146](#_Toc225064014)

[Meeting the “Access Control Polices Discretionary Access Control Policy Management Requirements” 150](#_Toc225064015)

[Addressing 2.1.2.1 “The OS shall allow only authorized administrators, object owners, and users with the DAC change authorization the ability to change the access permission associated with a named object” 150](#_Toc225064016)

[Addressing 2.1.2.2 “The OS shall allow only authorized administrators the ability to change object ownership” 150](#_Toc225064017)

[Addressing 2.1.2.3 “The OS shall provide only authorized administrators the ability to specify alternative initial values for default Discretionary Access Control object security attributes” 151](#_Toc225064018)

[Meeting the “Access Control Polices Discretionary Access Control Policy Audit Requirements” 152](#_Toc225064019)

[Addressing 2.1.3.1 “The OS shall provide the ability to audit all Discretionary Access Control policy decisions” 152](#_Toc225064020)

[Addressing 2.1.3.2 “The OS shall provide the ability to audit all changes to an object’s Discretionary Access Control security attributes” 155](#_Toc225064021)

[Addressing 2.1.3.3 “The OS shall provide the ability to audit all modifications to default Discretionary Access Control object security attribute values” 157](#_Toc225064022)

[Meeting the “Access Control Polices Mandatory Integrity Control Policy Functional Requirements” 158](#_Toc225064023)

[Addressing 2.2.1.1 “The OS shall enforce a Mandatory Integrity Control policy on untrusted subjects, named objects, and network interfaces and all operations that cause information to flow among them” 158](#_Toc225064024)

[Windows OS named objects subject to the MIC policy 158](#_Toc225064025)

[Windows OS Network List Manager (NLM) 159](#_Toc225064026)

[Addressing 2.2.1.2 “The OS shall associate an integrity label with each subject, named object, and network interface that accurately represents its integrity level” 159](#_Toc225064027)

[Windows OS integrity levels 159](#_Toc225064028)

[Windows OS algorithm to assign the integrity level to an access token for a Windows OS process 160](#_Toc225064029)

[MIC access operations on named objects under the MIC policy 161](#_Toc225064030)

[Content of a SYSTEM\_MANDATORY\_LABEL\_ACE ACE 162](#_Toc225064031)

[Resource Managers assigning SYSTEM\_MANDATORY\_LABEL\_ACE ACEs to named objects 163](#_Toc225064032)

[Addressing 2.2.1.3 “The OS shall enforce the Mandatory Integrity Control policy based on the specific types of the subject and object integrity attributes” 163](#_Toc225064033)

[Addressing 2.2.1.4 “The OS shall enforce specific relationships for any two valid integrity labels” 164](#_Toc225064034)

[Addressing 2.2.1.5 “The OS shall permit an information flow among subjects and objects based on a specific set of rules” 165](#_Toc225064035)

[Addressing 2.2.1.6 “The OS shall enforce that all definition of integrity labels are unique” 168](#_Toc225064036)

[Meeting the “Access Control Polices Mandatory Integrity Control Policy Management Requirements” 169](#_Toc225064037)

[Addressing 2.2.2.1 “The OS shall provide the ability to set the system-wide definition of integrity labels to authorized administrators” 169](#_Toc225064038)

[Addressing 2.2.2.2 “The OS shall provide authorized administrators the ability to change integrity labels of subjects and objects” 169](#_Toc225064039)

[Meeting the “Access Control Polices Mandatory Integrity Control Policy Audit Requirements” 170](#_Toc225064040)

[Addressing 2.2.3.1 “The OS shall provide the ability to audit all Mandatory Integrity Control policy decisions” 170](#_Toc225064041)

[Addressing 2.2.3.2 “The OS shall provide the ability to audit the setting and changing of system-wide integrity label definitions” 170](#_Toc225064042)

[Addressing 2.2.3.3 “The OS shall provide the ability to audit the changing of subject and object integrity labels” 170](#_Toc225064043)

[Meeting the “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Functional Requirements” 171](#_Toc225064044)

[Addressing 3.1.1.1 “The OS shall require each user to be uniquely identified and successfully authenticated by means of a password before allowing any actions on behalf of that user” 171](#_Toc225064045)

[Local user logging on 171](#_Toc225064046)

[Remote access attempt over the network interfaces 173](#_Toc225064047)

[Summarizing the arguments for meeting the requirement 174](#_Toc225064048)

[Addressing 3.1.1.2 “The OS shall be able to support passwords up to 32 characters in length, consisting of any combination of upper and lower case letters, numbers, and punctuations” 176](#_Toc225064049)

[Addressing 3.1.1.3 “The OS authentication mechanism shall provide a specific set of capabilities” 177](#_Toc225064050)

[Addressing 3.1.1.4 “The OS shall provide only obscured feedback to the user while the authentication is in progress” 178](#_Toc225064051)

[Addressing 3.1.1.5 “The OS shall detect when an authorized administrator specified positive integer of consecutive unsuccessful authentication attempts occur related to any authorized user authentication process” 178](#_Toc225064052)

[Addressing 3.1.1.6 “The OS shall perform a specific set of actions when the defined number of consecutive unsuccessful authentication attempts specified in “3.1.1.5” has been detected” 180](#_Toc225064053)

[Addressing 3.1.1.7 “The OS shall maintain a specific list of security attributes belonging to individual users” 182](#_Toc225064054)

[Addressing 3.1.1.8 “The OS shall associate a specific list of user security attributes with subjects acting on behalf of that user” 191](#_Toc225064055)

[Addressing 3.1.1.9 “The OS shall protect authentication data from disclosure through the use of operating system provided cryptographic services” 198](#_Toc225064056)

[Password lifecycle 199](#_Toc225064057)

[Addressing 3.1.1.10 “The OS shall provide an administrator role that is separate from untrusted users” 211](#_Toc225064058)

[Addressing 3.1.1.11 “The OS shall provide the authorized administrator the ability to set user attributes” 212](#_Toc225064059)

[Addressing 3.1.1.12 “The OS shall re-authenticate the user when changing authentication data” 215](#_Toc225064060)

[Addressing 3.1.1.13 “The OS shall provide a mechanism to verify that each user password when set meets authorized administrator configurable password characteristics” 215](#_Toc225064061)

[Addressing 3.1.1.14 “The OS shall automatically disable a user account when the current time has passed the authorized administrator specified time interval in which the user account has not been logged on successfully since the user account’s last successful logon time” 216](#_Toc225064062)

[Meeting the “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Management Requirements” 218](#_Toc225064063)

[Addressing 3.1.2.1 “The OS shall allow only authorized administrators the ability to create and manage user accounts” 218](#_Toc225064064)

[Addressing 3.1.2.2 “The OS shall provide authorized administrators the ability to specify a time interval in which the user account has not been logged on successfully so that a user account is automatically disabled after the time interval has elapsed” 218](#_Toc225064065)

[Addressing 3.1.2.3 “The OS shall allow only authorized administrators the ability to initially set and modify user security attributes (other than authentication data)” 219](#_Toc225064066)

[Addressing 3.1.2.4 “The OS shall allow only authorized administrators the ability to initialize user authentication data” 220](#_Toc225064067)

[Addressing 3.1.2.5 “The OS shall restrict the ability to modify authentication data to authorized administrators and users authorized to modify their own authentication data” 220](#_Toc225064068)

[Addressing 3.1.2.6 “The OS shall allow only authorized administrators the ability to initialize and modify authentication mechanism attributes” 221](#_Toc225064069)

[Addressing 3.1.2.7 “The OS shall provide authorized administrators the ability to specify and configure mandatory password composition” 225](#_Toc225064070)

[Meeting the “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Audit Requirements” 226](#_Toc225064071)

[Addressing 3.1.3.1 “The OS shall provide the ability to audit the creation and management of user accounts” 226](#_Toc225064072)

[Addressing 3.1.3.2 “The OS shall provide the ability to audit the initialization and modification of user security attributes” 228](#_Toc225064073)

[Addressing 3.1.3.3 “The OS shall provide the ability to audit the initialization and modification of authentication mechanism attributes” 231](#_Toc225064074)

[Addressing 3.1.3.4 “The OS shall provide the ability to audit when a user’s consecutive unsuccessful authentication attempts meets or exceeds the administrator-configured positive integer” 233](#_Toc225064075)

[Addressing 3.1.3.5 “The OS shall provide the ability to audit all user attempts to identify and authenticate to the system” 233](#_Toc225064076)

[Logon audit records generated at a Windows OS domain controller 233](#_Toc225064077)

[Logon audit records generated at an individual Windows OS machine 236](#_Toc225064078)

[Specific group assignment to a logged on user audit record 238](#_Toc225064079)

[Meeting the “Identification and Authentication User Interface Security Functional Requirements” 240](#_Toc225064080)

[Addressing 3.2.1.1 “Before establishing an interactive session, the OS shall display an authorized administrator specified advisory notice” 240](#_Toc225064081)

[Transitions from the “Welcome” state 241](#_Toc225064082)

[“Display legal notices” state 241](#_Toc225064083)

[Transitions from the “Display legal notices” state 241](#_Toc225064084)

[Addressing 3.2.1.2 “The OS shall allow user-initiated locking of an interactive session by performing a specific set of actions” 243](#_Toc225064085)

[Transitions from the “User logged on” state 244](#_Toc225064086)

[“Trusted path activated” state 244](#_Toc225064087)

[Transitions from the “Trusted path activated” state 245](#_Toc225064088)

[“Initiate lock on display areas” state 245](#_Toc225064089)

[Transitions from the “Initiate lock on display areas” state 246](#_Toc225064090)

[“Locked” state 246](#_Toc225064091)

[Transitions from the “Locked” state 247](#_Toc225064092)

[“Request to unlock” state 250](#_Toc225064093)

[Transitions from the “Request to unlock” state 250](#_Toc225064094)

[“User input inactivity timeout” policy 252](#_Toc225064095)

[“Screen saver” policy 253](#_Toc225064096)

[“Inactivity timeout handler during locked” state 254](#_Toc225064097)

[Transitions from the “Inactivity timeout handler during locked” state 255](#_Toc225064098)

[“Session disconnected during locked” state 257](#_Toc225064099)

[Transitions from the “Session disconnected during locked” state 258](#_Toc225064100)

[“Hibernate during locked” state 259](#_Toc225064101)

[Transitions from the “Hibernate during locked” state 260](#_Toc225064102)

[“Attempt to unlock” state 260](#_Toc225064103)

[Transitions from the “Attempt to unlock” state 261](#_Toc225064104)

[“Report unlock success” state 261](#_Toc225064105)

[Transitions from the “Report unlock success” state 261](#_Toc225064106)

[“Report unlock failure” state 262](#_Toc225064107)

[Transitions from the “Report unlock failure” state 266](#_Toc225064108)

[“Report last logon to user after unlock” state 269](#_Toc225064109)

[Transitions from the “Report last logon to user after unlock” state 270](#_Toc225064110)

[Summary of the Windows OS justification for meeting “3.2.1.2” 273](#_Toc225064111)

[Addressing 3.2.1.3 “The OS shall require a specific set of actions from users and administrators” 273](#_Toc225064112)

[“Switch user” button 274](#_Toc225064113)

[“Window logon state maintaining service user interactive application display area initialization” phase 276](#_Toc225064114)

[“Report logon service OK to logon UI service” state 277](#_Toc225064115)

[Transitions from the “Report logon service OK to logon UI service” state 278](#_Toc225064116)

[“Report last logon to user” state 278](#_Toc225064117)

[Transitions from the “Report last logon to user” state 279](#_Toc225064118)

[“Get session confirmation from LWTS” state 281](#_Toc225064119)

[Transitions from the “Get session confirmation from LWTS” state 286](#_Toc225064120)

[“Notify logon subscribers” state 291](#_Toc225064121)

[Transitions from the “Notify logon subscribers” state 294](#_Toc225064122)

[“Starting shell” state 296](#_Toc225064123)

[Transitions from the “Starting shell” state 296](#_Toc225064124)

[“User logged on” state 298](#_Toc225064125)

[Transitions from the “User logged on” state 302](#_Toc225064126)

[Summary of the Windows OS justification for meeting the requirement for an administrator to unlock and terminate a user session 306](#_Toc225064127)

[Addressing 3.2.1.4 “The OS shall enforce an authorized administrator specified maximum number of concurrent interactive sessions per user” 307](#_Toc225064128)

[Addressing 3.2.1.5 “The OS shall lock an interactive session after an authorized administrator specified time interval of user inactivity by performing a specific set of actions” 308](#_Toc225064129)

[“Inactivity timeout to run screensaver” state 309](#_Toc225064130)

[Transitions from the “Inactivity timeout to run screensaver” state 310](#_Toc225064131)

[“Post logged on inactivity timeout handler” state 311](#_Toc225064132)

[Transitions from the “Post logged on inactivity timeout handler” state 316](#_Toc225064133)

[“Post logged on resume” state 318](#_Toc225064134)

[Transitions from the “Post logged on resume” state 319](#_Toc225064135)

[“Initiate lock during resume” state 320](#_Toc225064136)

[Transitions from the “Initiate lock during resume” state 321](#_Toc225064137)

[“Locked during resume” state 321](#_Toc225064138)

[Transitions from the “Locked during resume” state 322](#_Toc225064139)

[“Inactivity timeout handler during locked” state 322](#_Toc225064140)

[Human interface device input and video output separation maintained by the Windows OS window manager(s) 322](#_Toc225064141)

[“Screen saver timeout” policy 323](#_Toc225064142)

[User input inactivity timeout notification from the Windows OS window manager 324](#_Toc225064143)

[Summary of the Windows OS justification for meeting “3.2.1.5” 324](#_Toc225064144)

[Addressing 3.2.1.6 “The OS shall provide the ability to deny interactive session establishment based on time and day” 324](#_Toc225064145)

[Addressing 3.2.1.7 “Upon successful interactive session establishment, the OS shall display to the authorized user a specific set of user access history information elements” 326](#_Toc225064146)

[Addressing 3.2.1.8 “The OS shall not erase the user access history information elements from the authorized user interface without giving the authorized user the opportunity to review the information elements” 328](#_Toc225064147)

[Meeting the “Identification and Authentication User Interface Security Management Requirements” 330](#_Toc225064148)

[Addressing 3.2.2.1 “The OS shall provide authorized administrators with the ability to specify an advisory notice” 330](#_Toc225064149)

[Addressing 3.2.2.2 “The OS shall provide authorized administrators with the ability to terminate an interactive session” 330](#_Toc225064150)

[Window terminal session specific “Window terminal session access” policy 330](#_Toc225064151)

[Capabilities available administrators 332](#_Toc225064152)

[Remote or local window terminal session disconnection 332](#_Toc225064153)

[Remote or local window terminal session reconnection from a (distinct) source window terminal session 334](#_Toc225064154)

[Forced logoff of a window terminal session 338](#_Toc225064155)

[Addressing 3.2.2.3 “The OS shall provide authorized administrators with the ability to specify the number of concurrent interactive sessions allowed per user” 339](#_Toc225064156)

[Addressing 3.2.2.4 “The OS shall provide authorized administrators with the ability to deny interactive session establishment based on system parameters specified in “3.2.1.6”” 340](#_Toc225064157)

[Addressing 3.2.2.5 “The OS shall provide authorized administrators with the ability to specify a time interval of session inactivity after which an (inactive) interactive session is locked” 341](#_Toc225064158)

[Meeting the “Identification and Authentication User Interface Security Audit Requirements” 342](#_Toc225064159)

[Addressing 3.2.3.1 “The OS shall provide the ability to audit all user attempts to re-authenticate to the system (for the interactive session unlocking purpose)” 342](#_Toc225064160)

[Addressing 3.2.3.2 “The OS shall provide the ability to audit all administrator attempts to terminate a user’s locked session” 342](#_Toc225064161)

[Addressing 3.2.3.3 “The OS shall provide the ability to audit any attempt to exceed the maximum number of concurrent interactive sessions by a user” 344](#_Toc225064162)

[Logging off audit event record generation 344](#_Toc225064163)

[Summary of the Windows OS justification for meeting “3.2.3.3” 344](#_Toc225064164)

[Addressing 3.2.3.4 “The OS shall provide the ability to audit all user attempts that violate the restrictions specified in “3.2.1.6”” 345](#_Toc225064165)

[Meeting the “Cryptographic Support Operations Functional Requirements” 347](#_Toc225064166)

[Addressing 4.1.1.1 “The OS shall provide a specific list of cryptographic services to applications” 347](#_Toc225064167)

[Meeting the “Cryptographic Support Operations Management Requirements” 348](#_Toc225064168)

[Meeting the “Cryptographic Support Operations Audit Requirements” 349](#_Toc225064169)

[Meeting the “Cryptographic Support Capabilities Functional Requirements” 350](#_Toc225064170)

[Addressing 4.2.1.1 “The OS shall provide FIPS-approved cryptographic algorithms and modes of operations, implemented in a crypto module that is FIPS 140-2 Security Level 2 validated” 350](#_Toc225064171)

[FIPS 140-2 “Cryptographic Module Specification” validation area 350](#_Toc225064172)

[FIPS 140-2 “Cryptographic Module Ports and Interfaces” validation area 351](#_Toc225064173)

[FIPS 140-2 “Roles, Services and Authentication” validation area 351](#_Toc225064174)

[FIPS 140-2 “Finite State Model” validation area 354](#_Toc225064175)

[FIPS 140-2 “Cryptographic Key Management” validation area 354](#_Toc225064176)

[FIPS 140-2 “Self-Tests” validation area 354](#_Toc225064177)

[FIPS 140-2 “Design Assurance” validation area 354](#_Toc225064178)

[Addressing 4.2.1.2 “A specific list of algorithms shall be provided for the corresponding cryptographic services” 355](#_Toc225064179)

[Encryption/Decryption 356](#_Toc225064180)

[Digital signature 357](#_Toc225064181)

[Hashing 358](#_Toc225064182)

[Key agreement 359](#_Toc225064183)

[Meeting the “Cryptographic Support Capabilities Management Requirements” 361](#_Toc225064184)

[Addressing 4.2.2.1 “The OS shall provide the authorized administrators the ability to select (when more than one algorithm is available) the appropriate algorithm for encryption/decryption, digital signature, hashing, and key agreement” 361](#_Toc225064185)

[Meeting the “Cryptographic Support Capabilities Audit Requirements” 362](#_Toc225064186)

[Meeting the “Cryptographic Support Randomization Functional Requirements” 363](#_Toc225064187)

[Addressing 4.3.1.1 “The OS crypto module shall provide Random Number Generation (RNG) services in accordance with a FIPS-Approved RNG listed in [FIPS 140-2 Annex C] composed of a specific method” 363](#_Toc225064188)

[Meeting the “Cryptographic Support Randomization Management Requirements” 366](#_Toc225064189)

[Meeting the “Cryptographic Support Randomization Audit Requirements” 367](#_Toc225064190)

[Meeting the “Cryptographic Support Cryptographic Key Management Functional Requirements” 368](#_Toc225064191)

[Addressing 4.4.1.1 “The OS crypto module shall generate symmetric cryptographic keys using a random number generator as specified in “4.3.1.1”” 368](#_Toc225064192)

[Addressing 4.4.1.2 “The OS crypto module shall generate asymmetric cryptographic keys using a domain parameter generator and a random number generator as specified in “4.3.1.1”” 368](#_Toc225064193)

[DSA algorithm 368](#_Toc225064194)

[rDSA algorithm 368](#_Toc225064195)

[ECDSA algorithm 369](#_Toc225064196)

[Diffie-Hellman Finite Field-based key agreement algorithm 370](#_Toc225064197)

[EC Diffie-Hellman Elliptic Curve-based key agreement algorithm 370](#_Toc225064198)

[Addressing 4.4.1.3 “The OS crypto module shall generate cryptographic keys such that generated key strength shall be equivalent to, or greater than, a symmetric key strength of 128 bits using conservative estimates as specified in [NIST SP 800-57 Section 5.6.1]” 371](#_Toc225064199)

[AES 372](#_Toc225064200)

[Diffie-Hellman Finite Field-based key agreement algorithm 372](#_Toc225064201)

[rDSA algorithm 373](#_Toc225064202)

[ECDSA algorithm 373](#_Toc225064203)

[EC Diffie-Hellman Elliptic Curve-based key agreement algorithm 374](#_Toc225064204)

[Addressing 4.4.1.4 “The OS crypto module shall perform key entry and output in accordance with the Level 1 Key Entry and Output requirements in [FIPS 140-2], “Security Requirements for Cryptographic Modules”” 375](#_Toc225064205)

[Addressing 4.4.1.5 “The OS crypto module shall provide the ability to protect secret keys, private keys, and critical cryptographic security parameters from unauthorized disclosure, modification, and substitution” 376](#_Toc225064206)

[Key isolation service 376](#_Toc225064207)

[Protections provided by the Windows OS key isolation service 389](#_Toc225064208)

[Addressing 4.4.1.6 “The OS crypto module shall provide the ability to protect public keys from unauthorized modification, and substitution” 392](#_Toc225064209)

[Addressing 4.4.1.7 “The OS crypto module shall provide the ability to correctly associate stored cryptographic keys (secret, private, or public) with the entity (e.g. person, group, or subject) to which the key is assigned” 395](#_Toc225064210)

[Addressing 4.4.1.8 “The OS crypto module shall destroy cryptographic keys in accordance with a specific cryptographic key zeroization method” 396](#_Toc225064211)

[Meeting the “Cryptographic Support Cryptographic Key Management Management Requirements” 398](#_Toc225064212)

[Meeting the “Cryptographic Support Cryptographic Key Management Audit Requirements” 399](#_Toc225064213)

[Addressing 4.4.3.1 “The OS shall provide the ability to audit attempts to violate the restrictions specified in “4.4.1.5”” 399](#_Toc225064214)

[Addressing 4.4.3.2 “The OS shall provide the ability to audit attempts to violate the restrictions specified in “4.4.1.6”” 401](#_Toc225064215)

[Meeting the “Cryptographic Support Cryptographic Testing Functional Requirements” 403](#_Toc225064216)

[Addressing 4.5.1.1 “The OS crypto module shall run a suite of self-tests to demonstrate the correct operation of the cryptographic functions in accordance with [FIPS 140-2 Section 4.9] during specific occasions” 403](#_Toc225064217)

[Windows OS Cryptographic Primitives Library (bcrypt.dll) 403](#_Toc225064218)

[Windows OS kernel security device driver (ksecdd.sys) 407](#_Toc225064219)

[Addressing 4.5.1.2 “If any of the cryptographic self-tests fail, the OS shall react as required by [FIPS 140-2 Section 4.9] for failing a self-test” 409](#_Toc225064220)

[Addressing 4.5.1.3 “The OS crypto module shall run a set of specific additional RNG tests” 411](#_Toc225064221)

[Meeting the “Cryptographic Support Cryptographic Testing Management Requirements” 415](#_Toc225064222)

[Addressing 4.5.2.1 “The OS shall provide the ability for an authorized administrator to run a suite of crypto module self-tests and RNG tests” 415](#_Toc225064223)

[Meeting the “Cryptographic Support Cryptographic Testing Audit Requirements” 416](#_Toc225064224)

[Addressing 4.5.3.1 “The OS shall audit failure of any crypto module self-tests and RNG tests” 416](#_Toc225064225)

[Meeting the “Information Protection Residual Information Functional Requirements” 418](#_Toc225064226)

[Addressing 5.1.1.1 “The OS shall ensure that any previous information content of a resource is made unavailable upon either the allocation of the resource to or the de-allocation of the resource from all non cryptographic objects” 418](#_Toc225064227)

[Local file system files 418](#_Toc225064228)

[Windows OS resource managers managing objects 419](#_Toc225064229)

[Windows OS microkernel managing frames 419](#_Toc225064230)

[Windows OS window terminal sessions 420](#_Toc225064231)

[Addressing 5.1.1.2 “Objects associated with cryptographic keys and critical cryptographic security parameters shall be destroyed as described in the Commercial Grade OS Requirement Set “4.4.1.8” requirement” 421](#_Toc225064232)

[Meeting the “Information Protection Residual Information Management Requirements” 424](#_Toc225064233)

[Meeting the “Information Protection Residual Information Audit Requirements” 425](#_Toc225064234)

[Meeting the “Information Protection Resource Control Functional Requirements” 426](#_Toc225064235)

[Addressing 5.2.1.1 “The OS shall enforce maximum quotas on the portion of shared persistent storage that individual authorized users can use” 426](#_Toc225064236)

[Meeting the “Information Protection Resource Control Management Requirements” 430](#_Toc225064237)

[Addressing 5.2.2.1 “The OS shall allow only authorized administrators the ability to set maximum quotas on shared persistent storage” 430](#_Toc225064238)

[Meeting the “Information Protection Resource Control Audit Requirements” 431](#_Toc225064239)

[Addressing 5.2.3.1 “The OS shall provide the ability to audit any request by a user that attempts to exceed the authorized administrator defined shared persistent storage quota” 431](#_Toc225064240)

[Addressing 5.2.3.2 “The OS shall provide the ability to audit the setting of maximum quotas on shared persistent storage” 432](#_Toc225064241)

[Meeting the “Information Protection Trusted Initialization~~, Self Testing and Recovery~~ Functional Requirements” 433](#_Toc225064242)

[~~Addressing 5.3.1.1 “The OS shall run a suite of self tests to demonstrate the correct operation of the OS security functions during specific occasions”~~ 433](#_Toc225064243)

[~~Addressing 5.3.1.2 “After a failure of a self test, the OS shall ensure a maintenance mode where the ability to return the OS to a secure state is provided”~~ 433](#_Toc225064244)

[Addressing 5.3.1.3 “The OS shall verify during initial startup the integrity of executable code that implements access control and cryptographic functionality through the use of the OS system provided cryptographic services” 433](#_Toc225064245)

[Integrity check conducted by the Windows OS boot manager 434](#_Toc225064246)

[Integrity check conducted by the Windows OS boot winload application 436](#_Toc225064247)

[Integrity check conducted by the Windows OS code integrity library module 437](#_Toc225064248)

[Chaining together the integrity checks 442](#_Toc225064249)

[Meeting the “Information Protection Trusted Initialization~~, Self Testing and Recovery~~ Management Requirements” 444](#_Toc225064250)

[~~Addressing 5.3.2.1 “The OS shall provide authorized administrators the ability to run the OS self tests”~~ 444](#_Toc225064251)

[~~Addressing 5.3.2.2 “The OS shall provide authorized administrators the ability to define the frequency at which self tests are automatically run”~~ 444](#_Toc225064252)

[Meeting the “Information Protection Trusted Initialization~~, Self Testing and Recovery~~ Audit Requirements” 445](#_Toc225064253)

[~~Addressing 5.3.3.1 “The OS shall provide the ability to audit any self test failures”~~ 445](#_Toc225064254)

[~~Addressing 5.3.3.2 “The OS shall provide the ability to audit when authorized administrators initiate the OS self tests”~~ 445](#_Toc225064255)

[Addressing 5.3.3.3 “The OS shall provide the ability to audit the failure of the (executable code) integrity verification” 445](#_Toc225064256)

[Meeting the “Import/Export of Data Exported Data Functional Requirements” 447](#_Toc225064257)

[Addressing 6.1.1.1 “The OS shall ensure that security attributes on named objects, when exported to removable media, are associated with the object” 447](#_Toc225064258)

[Meeting the “Import/Export of Data Exported Data Management Requirements” 449](#_Toc225064259)

[Meeting the “Import/Export of Data Exported Data Audit Requirements” 449](#_Toc225064260)

[Addressing 6.1.3.1 “The OS shall provide the ability to audit the exportation of named objects to removable media” 449](#_Toc225064261)

[Meeting the “Import/Export of Data Imported Data Functional Requirements” 451](#_Toc225064262)

[Addressing 6.2.1.1 “The OS shall enforce a specific set of rules when importing user data” 451](#_Toc225064263)

[Importing with the Data Decryption Field Data Recovery Field security attributes 451](#_Toc225064264)

[Importing without any security attribute 454](#_Toc225064265)

[Meeting the “Import/Export of Data Imported Data Management Requirements” 457](#_Toc225064266)

[Addressing 6.2.2.1 “The OS shall provide authorized administrators with the ability to select the technique for handling imported data with attributes that cannot be validated” 457](#_Toc225064267)

[Meeting the “Import/Export of Data Imported Data Audit Requirements” 458](#_Toc225064268)

[Addressing 6.2.3.1 “The OS shall provide the ability to audit the failure of security attribute validation” 458](#_Toc225064269)

[Addressing 6.2.3.2 “The OS shall provide the ability to audit the administrator selected configuration for handling imported data with attributes that cannot be validated” 458](#_Toc225064270)

[Meeting the “Import/Export of Data Trusted Channels Functional Requirements” 459](#_Toc225064271)

[Addressing 6.3.1.1 “The OS shall provide a communication path between itself and users that is logically distinct from other communication paths and provides assured identification of the OS to the requesting user” 459](#_Toc225064272)

[Addressing 6.3.1.2 “The OS shall permit users to initiate communication via a trusted path” 465](#_Toc225064273)

[Addressing 6.3.1.3 “The OS shall require the use of a trusted path for all user operations involving authentication data” 466](#_Toc225064274)

[Meeting the “Import/Export of Data Trusted Channels Management Requirements” 468](#_Toc225064275)

[Meeting the “Import/Export of Data Trusted Channels Audit Requirements” 469](#_Toc225064276)

[Meeting the “Revocation Access Revocation Functional Requirements” 470](#_Toc225064277)

[Addressing 7.1.1.1 “The OS shall enforce the revocation of security relevant attributes associated with named objects when access checks are made” 470](#_Toc225064278)

[Addressing 7.1.1.2 “The OS shall enforce the revocation of security relevant attributes associated with users at user session establishment” 470](#_Toc225064279)

[Addressing 7.1.1.3 “The OS shall immediately terminate all subjects associated with deleted user accounts” 475](#_Toc225064280)

[Meeting the “Revocation Access Revocation Management Requirements” 478](#_Toc225064281)

[Addressing 7.1.2.1 “The OS shall allow authorized administrators, object owners, and users with the DAC change attribute the ability to revoke security attributes associated with Discretionary Access Control policies on named objects” 478](#_Toc225064282)

[Addressing 7.1.2.2 “The OS shall allow authorized administrators and subjects with the MIC change attribute the ability to revoke security attributes associated with Mandatory Integrity Control policies on named objects” 478](#_Toc225064283)

[Addressing 7.1.2.3 “The OS shall allow authorized administrators the ability to revoke security attributes associated with users” 479](#_Toc225064284)

[Addressing 7.1.2.4 “The OS shall allow authorized administrators the ability to delete user accounts” 480](#_Toc225064285)

[Meeting the “Revocation Access Revocation Audit Requirements” 482](#_Toc225064286)

[Addressing 7.1.3.1 “The OS shall provide the ability to audit the revocation of security attribute” 482](#_Toc225064287)

[Meeting the “Auditing Audit Collection Functional Requirements” 484](#_Toc225064288)

[Addressing 8.1.1.1 “The OS shall be able to generate audit records for all security-relevant events identified in this Commercial Grade OS Requirement Set and the other specific security relevant auditable events designed to be generated by the OS claiming compliance with this Commercial Grade OS Requirement Set” 484](#_Toc225064289)

[Addressing 8.1.1.2 “The OS shall be able to associate each auditable event with the identity or the user that caused the event” 484](#_Toc225064290)

[Addressing 8.1.1.3 “The OS shall be able to monitor and report the accumulation of specific sets of audit events known to indicate a potential security violation and immediately report the accumulated events when a threshold is exceeded” 484](#_Toc225064291)

[Addressing 8.1.1.4 “The OS shall record, within each audit record, the specific set of information items, as appropriate to the audit event” 486](#_Toc225064292)

[Meeting the “Auditing Audit Collection Management Requirements” 488](#_Toc225064293)

[Addressing 8.1.2.1 “The OS shall provide an authorized administrator with the capability to manage the threshold values specified in “8.1.1.3”” 488](#_Toc225064294)

[Addressing 8.1.2.2 “The OS shall allow an authorized administrator to specify which events, from the set of auditable events, are to be audited” 489](#_Toc225064295)

[Meeting the “Auditing Audit Collection Audit Requirements” 498](#_Toc225064296)

[Addressing 8.1.3.1 “The OS shall provide the ability to audit the modification of the threshold values specified in “8.1.1.3”” 498](#_Toc225064297)

[Meeting the “Auditing Audit Storage Functional Requirements” 499](#_Toc225064298)

[Addressing 8.2.1.1 “The OS shall prevent modification of previously written audit records” 499](#_Toc225064299)

[Windows OS audit policy unit 499](#_Toc225064300)

[Windows OS event tracing (ETW) facility 499](#_Toc225064301)

[“Etw Logger” Windows OS system thread 501](#_Toc225064302)

[Windows OS security audit store service 501](#_Toc225064303)

[“Eventlog-Security” ETW session establishment 502](#_Toc225064304)

[Windows OS audit policy unit submitting audit records directly to the Windows OS security audit store service “Security” channel 503](#_Toc225064305)

[Persistent security log file 504](#_Toc225064306)

[Audit records written by the Windows OS security audit store service 505](#_Toc225064307)

[Prevention of previously generated audit record modification 506](#_Toc225064308)

[Addressing 8.2.1.2 “The OS shall provide the capability for authorized administrators to specify the specific actions to be taken upon audit storage exhaustion” 507](#_Toc225064309)

[Meeting the “Auditing Audit Storage Management Requirements” 510](#_Toc225064310)

[Addressing 8.2.2.1 “The OS shall provide an authorized administrator with the capability to specify actions to be taken upon audit storage exhaustion” 510](#_Toc225064311)

[Addressing 8.2.2.2 “The OS shall provide an authorized administrator with the capability to sort, select and review collected audit records based on identity and any audit information items specified in “8.1.1.4”” 510](#_Toc225064312)

[Addressing 8.2.2.3 “The OS shall provide an authorized administrator with the capability to archive audit data” 511](#_Toc225064313)

[Meeting the “Auditing Audit Storage Audit Requirements” 513](#_Toc225064314)

[Addressing 8.2.3.1 “The OS shall provide the ability to audit the deleting or archiving of audit data” 513](#_Toc225064315)

[Appendix A: Administrative Privileged Application Launching Service 514](#_Toc225064316)

[Appendix B: Addressing Certain Interesting Security Functional Requirements 527](#_Toc225064317)

[Meeting Additional “Security Architecture Separation, Isolation, and Least Privilege Functional Requirements” 528](#_Toc225064318)

[Addressing 1.1.1.6 “The OS Memory protection shall provide specific capabilities” 528](#_Toc225064319)

[Addressing 1.1.1.7 “Separation of operating system programs (execution contexts) from user programs should be realized in the OS” 529](#_Toc225064320)

[Addressing 1.1.1.8 “In a single-user system, the OS memory protection shall prevent user subject processes from affecting the OS” 529](#_Toc225064321)

[Addressing 1.1.1.9 “In a multi-user system, the OS memory protection shall prevent the interference between user subjects within the OS” 529](#_Toc225064322)

[Addressing 1.1.1.10 “System “back door” shall not be designed in the OS” 529](#_Toc225064323)

[Meeting Additional “Security Architecture Distributed Architectures Functional Requirements” 531](#_Toc225064324)

[Addressing 1.2.1.6 “The OS shall achieve the basic protection to security-relevant data transmission, data separation transmission and data integrity protection within the OS” 531](#_Toc225064325)

[Meeting Additional “Security Architecture Distributed Architectures Management Requirements” 532](#_Toc225064326)

[Addressing 1.2.2.2 “The OS shall achieve the protection to usability, confidentiality, and integrity of output security-relevant data” 532](#_Toc225064327)

[Addressing 1.2.2.3 “The OS shall provide an installation mechanism for the setting and upgrade of configuration parameters” 532](#_Toc225064328)

[Addressing 1.2.2.4 “Before the initialization and the implementation of protection to security-relevant data structure, security policy attributes of users and administrators should be defined” 532](#_Toc225064329)

[Addressing 1.2.2.5 “After the completion of OS installation and before the access by ordinary users, the OS should configure responsibilities of initial users and administrators, root directory, audit parameters, system audit trail setup, and appropriate access control on files and directories” 533](#_Toc225064330)

[Addressing 1.2.2.6 “Application programs provided by the OS should (by default) be limited to the effective use of the OS, and only administrators can modify or replace application programs provided by the OS” 533](#_Toc225064331)

[Addressing 1.2.2.7 “The OS shall provide a mechanism for security administrator of operating system to generate detailed report on security parameters” 534](#_Toc225064332)

[Addressing 1.2.2.8 “The OS environment shall be able to control the use of the OS control panel” 534](#_Toc225064333)

[Meeting Additional “Security Architecture Distributed Architectures Audit Requirements” 537](#_Toc225064334)

[Addressing 1.2.3.2 “The OS environment shall be able to audit the use of system control panel” 537](#_Toc225064335)

[Meeting Additional “Access Control Polices Discretionary Access Control Policy Management Requirements” 538](#_Toc225064336)

[Addressing 2.1.2.4 “Owner of an object shall be the only subject with the right to modify the access right to the object” 538](#_Toc225064337)

[Addressing 2.1.2.5 “Distribution of the control rights on an object to other subjects by the object’s owner shall be forbidden” 538](#_Toc225064338)

[Meeting Additional “Access Control Polices Mandatory Integrity Control Policy Functional Requirements” 540](#_Toc225064339)

[Addressing 2.2.1.7 “Function on the protection of integrity of user data transmitted within the OS, such as inter-process communication data shall be provided: Integrity label shall be moved with data” 540](#_Toc225064340)

[Addressing 2.2.1.8 “Function on the protection of integrity of user data transmitted within the OS, such as inter-process communication data shall be provided: The system shall ensure that data with lower integrity cannot be inserted into and cannot replace the data with higher integrity” 542](#_Toc225064341)

[Addressing 2.2.1.9 “MIC based integrity protection of cross-network user data residing in file objects shall be realized by taking into account of the consistency of security attributes between subjects and file/directory objects of each OS computer in a network environment” 542](#_Toc225064342)

[Meeting the “Access Control Polices Mandatory Access Control Policy for User File/Directory Objects Functional Requirements” 544](#_Toc225064343)

[Addressing 2.3.1.1 “The OS shall adopt a labeling method to set sensitive labels for subjects and user file/directory objects” 544](#_Toc225064344)

[Poor man’s Mandatory Access Control (MAC) for user file and directory objects 544](#_Toc225064345)

[Addressing 2.3.1.2 “The sensitive labels constitute an attribute repository with a multi-level security model for user file/directory objects” 547](#_Toc225064346)

[Addressing 2.3.1.3 “The OS shall enforce a Mandatory Access Control policy on untrusted subjects, named user file/directory objects and all operations that cause information to flow among them” 548](#_Toc225064347)

[Addressing 2.3.1.4 “A uniform mandatory access control function for user file/directory objects shall be provided for the distributed OS running in a network environment” 548](#_Toc225064348)

[Addressing 2.3.1.5 “MAC based confidentiality protection of cross-network user data residing in file objects shall be realized by taking into account of the consistency of security attributes between subjects and file/directory objects of each OS computer in a network environment” 549](#_Toc225064349)

[Addressing 2.3.1.6 “The OS shall permit an information flow among subjects and user file/directory objects based on a specific set of rules based on hierarchical sensitivity levels” 549](#_Toc225064350)

[Meeting the “Access Control Polices Mandatory Access Control Policy for User File/Directory Objects Management Requirements” 551](#_Toc225064351)

[Addressing 2.3.2.1 “The sensitive labels for user file/directory objects and subjects of operating system should be established, maintained, and managed by authorized administrators” 551](#_Toc225064352)

[Meeting the “Access Control Polices Mandatory Access Control Policy for User File/Directory Objects Audit Requirements” 552](#_Toc225064353)

[Addressing 2.3.3.1 “The OS shall provide the ability to audit all Mandatory Access Control policy decisions” 552](#_Toc225064354)

[Addressing 2.3.3.2 “The OS shall provide the ability to audit the changing of subject and object sensitivity labels” 552](#_Toc225064355)

[Meeting Additional “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Functional Requirements” 553](#_Toc225064356)

[Addressing 3.1.1.15 “User authentication shall adopt a selection of password authentication, token based dynamic password authentication, biometric authentication, digital certificate authentication and others” 553](#_Toc225064357)

[Addressing 3.1.1.16 “Before the establishment of an OS logon session, user identity shall be authenticated and the logon mechanism is not allowed to bypass the authentication mechanism” 553](#_Toc225064358)

[Addressing 3.1.1.17 “The OS shall realize authentication failure treatment through pre-definition of unsuccessful trial values for authentication (including the time threshold value)” 554](#_Toc225064359)

[Addressing 3.1.1.18 “The OS shall realize provision of treatment measures when the time threshold value (of 3.1.1.17) is reached” 554](#_Toc225064360)

[Addressing 3.1.1.19 “The OS shall realize authentication failure treatment through pre-definition of unsuccessful trial values for authentication (including number of trial times)” 554](#_Toc225064361)

[Addressing 3.1.1.20 “The OS shall realize provision of treatment measures when the number of trial times (of 3.1.1.19) is reached” 554](#_Toc225064362)

[Addressing 3.1.1.21 “The OS shall realize unique user identifier in the whole life cycle of the OS, and the consistency between user names or alternative names and UIDs when User name and user identifier (UID) are adopted by the OS user identification” 555](#_Toc225064363)

[Addressing 3.1.1.22 “User subject binding function shall be designed and realized for users, who register with the OS by connecting user process with the owner, to ensure that the behavior of the user process can be traced back to all owners of the process” 556](#_Toc225064364)

[Addressing 3.1.1.23 “User subject binding function shall be designed and realized for users, who register with the OS by dynamically connecting the system process with the current service requester, to ensure that the behavior of the system process can be traced back to current service requester” 556](#_Toc225064365)

[Meeting Additional “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Audit Requirements” 558](#_Toc225064366)

[Addressing 3.1.3.6 “The OS shall point out the least types of recordable audit events, including the change of system database management -- distribution and setup of users ID” 558](#_Toc225064367)

[Addressing 3.1.3.7 “The OS shall point out the least types of recordable audit events, including the change of system database management -- the change of user account attributes” 558](#_Toc225064368)

[Addressing 3.1.3.8 “The OS shall point out the least types of recordable audit events, including the change of user identities as the order of super-users” 558](#_Toc225064369)

[Addressing 3.1.3.9 “The OS shall point out the least types of recordable audit events, including the success and failure of login for session establishment, operational system interfaces” 559](#_Toc225064370)

[Meeting Additional “Identification and Authentication User Interface Security Functional Requirements” 560](#_Toc225064371)

[Addressing 3.2.1.9 “The OS shall provide a mechanism for locking user keyboard, and request user authentication for the keyboard unlocking process” 560](#_Toc225064372)

[Addressing 3.2.1.10 “The OS shall be able to limit the maximum quantity of concurrent sessions of system, and use default values as the limited number of quantity of session, based on the basic user identification” 560](#_Toc225064373)

[Addressing 3.2.1.11 “After exceeding the set unused time limit, the OS shall disconnect the session or re-authenticate the user, and the system shall provide a default value on time limit” 561](#_Toc225064374)

[Addressing 3.2.1.12 “The OS shall provide a mechanism that can regulate who can access to the system in accordance with time, access methods, location, network address or ports, etc” 561](#_Toc225064375)

[Addressing 3.2.1.13 “After the successful login, the OS shall record and show users specific information items” 562](#_Toc225064376)

[Addressing 3.2.1.14 “The OS shall be able enforce different password restriction and accout lockout policies in a user’s logging on authentication attempts based on different roles of the user” 563](#_Toc225064377)

[Meeting Additional “Information Protection Residual Information Functional Requirements” 567](#_Toc225064378)

[Addressing 5.1.1.3 “The OS shall ensure the resources under dynamic allocation and management can be re-used, when the information security is maintained, through specific actions” 567](#_Toc225064379)

[Meeting Additional “Information Protection Resource Control Functional Requirements” 568](#_Toc225064380)

[Addressing 5.2.1.2 “The OS shall ensure its normal running by some measures when the OS is under identified failure (i.e. degraded) conditions, where the OS shall check and report when the OS service level is lower than the predetermined minimum” 568](#_Toc225064381)

[Windows OS Reliability and Performance Monitor 568](#_Toc225064382)

[Addressing 5.2.1.3 “The OS shall detect and send out reports to registered administrators when the OS service level of system resources is lower than the predetermined minimum” 589](#_Toc225064383)

[Addressing 5.2.1.4 “The OS shall adopt appropriate policies for providing the priority to the subject for the use of a resource subset within the OS based on the limited service priority” 590](#_Toc225064384)

[Windows OS Priority Scheduling Scheme 590](#_Toc225064385)

[Addressing 5.2.1.5 “The OS shall implement the management and allocation of the OS resources in accordance with requirements of ceiling on resources allocation, and make sure that users/subjects will not monopolize any managed resource by implementing the quota system” 592](#_Toc225064386)

[Windows OS Quota Control for CPU Rate and Memory Usage 592](#_Toc225064387)

[Addressing 5.2.1.6 “The OS shall ensure resources can be accessed and utilized when an authorized subject send out request” 595](#_Toc225064388)

[Addressing 5.2.1.7 “The OS shall provide a mechanism based on one user or one user group to and control the consumption of disk” 595](#_Toc225064389)

[Addressing 5.2.1.8 “The OS shall provide a mechanism based on one user or one user group to and control the utilization of CPU” 595](#_Toc225064390)

[Meeting Additional “Information Protection Resource Control Management Requirements” 596](#_Toc225064391)

[Addressing 5.2.2.2 “The OS shall implement the management and allocation of the OS resources” 596](#_Toc225064392)

[Meeting Additional “Information Protection Resource Control Audit Requirements” 597](#_Toc225064393)

[Addressing 5.2.3.3 “The OS shall point out the least types of recordable audit events, including the operations of system administrator and system security administrator and others” 597](#_Toc225064394)

[Addressing 5.2.3.4 “The OS shall point out the least types of recordable audit events, including the deletion of objects” 597](#_Toc225064395)

[Meeting Additional “Information Protection Self Testing Functional Requirements” 599](#_Toc225064396)

[Addressing 5.3.1.4 “The OS shall provide the mechanisms and processes for regular confirmation of correct OS operations” 599](#_Toc225064397)

[Meeting the “Information Protection Persistent Storage Data Availability Functional Requirements” 600](#_Toc225064398)

[Addressing 5.4.1.1 “While handling data access, the OS shall check whether requested data on storage medium has corruption errors, and restore the data if a corruption error is detected” 600](#_Toc225064399)

[Self healing Windows OS NTFS 600](#_Toc225064400)

[Addressing 5.4.1.2 “The OS shall use an integrity function provided by the encryption support system for the integrity protection of encrypted and stored data” 611](#_Toc225064401)

[Addressing 5.4.1.3 “To the data of the OS stored on disk, the OS shall realize an “automatically check the integrity of files” function” 612](#_Toc225064402)

[Addressing 5.4.1.4 “To the data of the OS stored on disk, the OS shall realize an “automatically check the integrity of disk surface” function” 613](#_Toc225064403)

[Addressing 5.4.1.5 “To the data of the OS stored on disk, the OS shall realize a “check and diagnose as well as recover from errors on the disk at any time” function” 614](#_Toc225064404)

[Addressing 5.4.1.6 “To the data of the OS stored on disk, the OS shall realize a “repair sector intertwining and sectors lost” function” 614](#_Toc225064405)

[Addressing 5.4.1.7 “To the data of the OS stored on disk, the OS shall realize a “move data to good sectors” function” 615](#_Toc225064406)

[Addressing 5.4.1.8 “To the data of the OS stored on disk, the OS shall realize a “hard disk data backup and restoration” function” 615](#_Toc225064407)

[Windows OS Backup and Restore 615](#_Toc225064408)

[Addressing 5.4.1.9 “To the data of the OS stored on disk, the OS shall realize a “data compression on hard disk” function” 623](#_Toc225064409)

[Addressing 5.4.1.10 “To the data of the OS stored on disk, the OS shall realize a “backup on hard disk and restore if necessary” function” 623](#_Toc225064410)

[Addressing 5.4.1.11 “For data to be stored on disk, the OS data-writing operation shall be designed with an “invert back” requirement, where the data-writing operation can be inverted back in case of abnormalities” 623](#_Toc225064411)

[Addressing 5.4.1.12 “The OS shall provide a utility program for inspecting the integrity of file systems and disks and the utility program can be operated automatically by the OS” 625](#_Toc225064412)

[Addressing 5.4.1.13 “In case of the failure or interruption of the OS, a restoration of the OS with the minimum amount of damage is available” 626](#_Toc225064413)

[Addressing 5.4.1.14 “The OS shall provide the processes for software and data backup and restoration, where a synchronization point for restart is added to a backup instance for a subsequent restoration” 626](#_Toc225064414)

[Meeting the “Information Protection Persistent Storage Data Availability Management Requirements” 627](#_Toc225064415)

[Meeting the “Information Protection Persistent Storage Data Availability Audit Requirements” 628](#_Toc225064416)

[Addressing 5.4.3.1 “To the data of the OS stored on disk, the OS shall realize an “automatically record problems of disk surface” function” 628](#_Toc225064417)

[Meeting the “Information Protection Maintenance Mode Access Control Functional Requirements” 629](#_Toc225064418)

[Addressing 5.5.1.1 “The ordinary operation mode and system maintenance mode of the OS should be differentiated” 629](#_Toc225064419)

[Characterizing the Windows OS “Maintenance Modes” 629](#_Toc225064420)

[Addressing 5.5.1.2 “The OS shall prevent an ordinary user from entering the maintenance mode from unauthorized system” 637](#_Toc225064421)

[Addressing 5.5.1.3 “For backup or conventional system maintenance tasks which do not affect the security subsystem of the OS, not all these system maintenance tasks are carried out under the maintenance mode” 637](#_Toc225064422)

[Addressing 5.5.1.4 “After the completion of the OS installation and before the access by ordinary users, the OS shall configure responsibilities of initial users and administrators, root directory, audit parameters, system audit trail setup, and appropriate access control on files and directories” 638](#_Toc225064423)

[Addressing 5.5.1.5 “The OS shall have the ability to run under the maintenance mode, and the OS can only be used by system administrators under the maintenance mode when specific security functions fail” 638](#_Toc225064424)

[Meeting the “Information Protection Maintenance Mode Access Control Management Requirements” 639](#_Toc225064425)

[Meeting the “Information Protection Maintenance Mode Access Control Audit Requirements” 640](#_Toc225064426)

[Meeting Additional “Import/Export of Data Exported Data Functional Requirements” 641](#_Toc225064427)

[Addressing 6.1.1.2 “When information is sent from the scope under the OS control to the scope beyond the OS control, sensitive labels can be attached or not attached” 641](#_Toc225064428)

[Meeting Additional “Import/Export of Data Imported Data Functional Requirements” 642](#_Toc225064429)

[Addressing 6.2.1.2 “When information is sent from the scope beyond the OS control to the scope under the OS control, sensitive labels should be attached” 642](#_Toc225064430)

[Meeting Additional “Auditing Audit Collection Functional Requirements” 644](#_Toc225064431)

[Addressing 8.1.1.5 “Security audit function shall be closely combined with authentication, discretionary access control, labeling, mandatory access control, integrity control and other security functions” 644](#_Toc225064432)

[Addressing 8.1.1.6 “The OS shall point out the least types of recordable audit events, including the change of system database management -- the addition or change of system procedures or processes” 644](#_Toc225064433)

[Addressing 8.1.1.7 “The OS shall point out the least types of recordable audit events, including the change of system database management -- the change of date and time” 645](#_Toc225064434)

[Addressing 8.1.1.8 “Data records of each (audit) event shall include specific information” 646](#_Toc225064435)

[Addressing 8.1.1.9 “The OS shall provide audit logs, real-time alarm generation, potential security risk analysis, basic abnormality inspection” 647](#_Toc225064436)

[Addressing 8.1.1.10 “The OS shall provide the history records on modification of system resources that can be accessed by users” 647](#_Toc225064437)

[Meeting Additional “Auditing Audit Collection Management Requirements” 649](#_Toc225064438)

[Addressing 8.1.2.3 “The OS shall provide security audit event selection” 649](#_Toc225064439)

[Addressing 8.1.2.4 “The OS shall point out the least types of recordable audit events, including the change of system database management -- audit trail setup and analysis” 649](#_Toc225064440)

[Addressing 8.1.2.5 “The OS shall provide a protected mechanism for opening and closing audit” 649](#_Toc225064441)

[Addressing 8.1.2.6 “The protection to function related to audit trail and management shall be provided in order to fulfill the creation, destruction, emptying and archiving of audit trail” 650](#_Toc225064442)

[Addressing 8.1.2.7 “A mechanism shall be provided to show the current [audit policy] selection, and users of the [show the current [audit policy] selection] mechanism should be limited [to the] authorized users” 650](#_Toc225064443)

[Meeting Additional “Auditing Audit Collection Audit Requirements” 652](#_Toc225064444)

[Addressing 8.1.3.2 “The OS shall point out the least types of recordable audit events, including the operations of system administrator and system security administrator and others” 652](#_Toc225064445)

[Meeting Additional “Auditing Audit Storage Functional Requirements” 653](#_Toc225064446)

[Addressing 8.2.1.3 “The OS shall provide protected audit trail record” 653](#_Toc225064447)

[Addressing 8.2.1.4 “The audit process shall be generated, maintained and protected to avoid modification, unauthorized access and damage” 653](#_Toc225064448)

[Addressing 8.2.1.5 “The OS shall provide protection on audit records from being modified and destroyed by unauthorized users” 653](#_Toc225064449)

[Addressing 8.2.1.6 “When the audit is activated, the integrity of audit trail events shall be guaranteed (i.e. maintained)” 653](#_Toc225064450)

[Addressing 8.2.1.7 “Audit data which requires special protection shall be protected” 654](#_Toc225064451)

[Addressing 8.2.1.8 “Administrators should be able to define the thresholds beyond the limit of audit trail” 654](#_Toc225064452)

[Meeting Additional “Auditing Audit Storage Management Requirements” 655](#_Toc225064453)

[Addressing 8.2.2.4 “The OS shall provide basic audit review, and limited audit review, optional audit inquiry” 655](#_Toc225064454)

[Addressing 8.2.2.5 “The access of unauthorized users to audit data shall be strictly restricted” 655](#_Toc225064455)

[Addressing 8.2.2.6 “Protection on audit records from being accessed by unauthorized users shall be provided” 655](#_Toc225064456)

[Addressing 8.2.2.7 “Audit tools should be able to be authorized to an individual for supervision and browsing audit data” 655](#_Toc225064457)

# Meeting the “Security Architecture Separation, Isolation, and Least Privilege Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 5 individual functional requirements under the heading of “Security Architecture Separation, Isolation, and Least Privilege Functional Requirements”. They are listed as “1.1.1.n”, where n = 1, 2, 3, 4, and 5.

## Addressing 1.1.1.1 “The OS shall protect security-relevant data from unauthorized access”

Based on the concept of “defense in depth”, this specific functional requirement can be addressed at multiple levels of protection. These levels of protection are:

* Full volume encryption;
* User authentication;
* User mode kernel mode context switching;
* Process memory virtualization;
* Window terminal session isolation;
* Secure desktop separation;
* Isolation of user-entered credential through encryption;
* User interface privilege isolation within the same desktop;
* Persistent data access mediation;
* Encrypting file system;
* Long lived cryptographic key isolation service.

This view of the Windows OS through different levels is necessary, because an operating system typically may be viewed at varying levels of abstraction.



### Full volume encryption

The first level of protection is the full volume encryption (FVE) for the hard disk volume, where the Windows operating system (Windows OS) binary files and their associated system data reside in. Before the Windows OS binary files could be decrypted so that the Windows OS would boot from these files, the owner user of the machine must provide a valid FVE PIN through the attached keyboard or insert a USB drive which stores a valid FVE PIN. In other words, an unauthorized user, without possession of the valid FVE PIN, has access to only the Windows OS binary files in the encrypted form.

If the Windows OS binary files are deemed as security-relevant data, then the FVE relies on the valid FVE PIN to protect the clear-text Windows OS binary files from unauthorized access.

### User authentication

The second level of protection occurs after the Windows OS has booted. This protection is the user authentication. It is in action when a user locally or remotely attempts to gain access to protected resources or services through interfaces implemented or supported by the Windows OS. The user is represented by a unique security ID (SID) upon the successful user authentication. Every Windows OS process or thread created by the Windows OS process manager is assigned the association with a user SID, which is used for subsequent user identification purpose. The Windows OS process manager exists in the kernel mode. Due to the user mode kernel mode isolation (which is described next), its operations on processes and threads are not interfered by relatively untrusted code, which exists only in the user mode. Unsuccessful user authentication does not result in a Windows OS process or thread being associated with a user SID, which can identify a specific user account.

Assume that certain security-relevant data is restricted, where an input user SID is required for the associated access control decision. Then a failed user authentication protects this security-relevant data from unauthorized access.

### User mode kernel mode context switching

The third level of protection is the user mode kernel mode context switching. Because the Windows OS memory manager maps kernel mode (i.e. trusted) code into every process, it is necessary that the kernel mode code is isolated from the user mode code, which is potentially untrusted. This isolation is accomplished by the user mode kernel mode context switching, which relies on the use of specific instructions of the underlying hardware processor architecture to let the processor architecture reset its current privilege level. The kernel mode (i.e. trusted) code is executed only when the current privilege level of the processor architecture is at the highest. The user mode code is executed only when the current privilege level of the processor architecture is at the lowest.

When the user mode code of a Windows OS process or thread attempts to gain access to kernel mode resources, it is necessary to enter specific kernel mode interfaces of the Windows OS, which conduct the user mode kernel mode context switching. After the context switching, the corresponding kernel mode code takes over the execution control. When the kernel mode code finishes its executions, it gives up its execution control so that the original context, which was maintained just before the context switching, is restored.

As kernel mode resources are deemed as security-relevant data and the enforcement for their access occurs in the kernel mode code, these kernel mode resources are protected from unauthorized access.

The Windows OS kernel mode interfaces are not meant to be by-passed due to the processor architecture. Certain processor architecture instructions, which could be used to deny the processor service to other processes, abuse or monitor the processor performance, or subvert the processor memory protection mechanisms, are deemed as privileged instructions. The processor architecture ensures that the use of these privileged instructions is limited to code (i.e. kernel mode code) which is executed only when the current privilege level of the processor architecture is at the highest. In addition, the Windows OS also relies on the physical memory page protection mechanisms of the processor as follows.

If the Windows OS marks a page that is limited to code (i.e. kernel mode code) which is executed only when the current privilege level of the processor architecture is at the highest, then that page is not accessible from the user mode code. This page is deemed as security-relevant data. Therefore, it is protected from unauthorized access.

The Windows OS can also mark a page as “read-only”. A page-fault exception occurs when there is an attempt to write to a “read-only” page. This page is deemed as security-relevant data. Therefore, it is protected from unauthorized access.

The physical memory page protection mechanisms of the processoralso allow the Windows OS memory manager to mark one or more pages of memory as non-executable. Marking memory regions as non-executable means that code cannot be run from that region of memory, which makes it harder for the exploitation of buffer overruns. If a subject, running in a Windows OS process, attempts to run code from a data page that is protected, a memory access violation exception occurs. Furthermore, if the exception is not handled, the Windows OS process is automatically terminated. As the execute-ability of a “non-executable” marked page may be deemed as security-relevant, the page is protected from unauthorized access also. The non-execute-ability of a “non-executable” marked page is the corner stone of the Windows OS [Data Execution Protection (DEP)](http://msdn.microsoft.com/en-us/library/aa366553.aspx). The [system wide DEP policy](http://msdn.microsoft.com/en-us/library/bb736298(VS.85).aspx) content is read during boot. It is defined in [the corresponding Boot Configuration Data (BCD) settings for DEP](http://msdn.microsoft.com/en-us/library/aa362671(VS.85).aspx) as follows.

* AlwaysOff (0)
  + DEP is disabled for all parts of the system, regardless of hardware processorsupport for DEP. The processor runs in the [Physical Address Extension (PAE)](http://msdn.microsoft.com/en-us/library/aa366796.aspx) mode with 32-bit versions of Windows unless PAE is disabled in the boot configuration data.
* AlwaysOn (1)
  + DEP is enabled for all parts of the system. All processes always run with DEP enabled. DEP cannot be explicitly disabled for selected applications. System compatibility fixes are ignored.
* OptIn (2)
  + On systems with processors that are capable of hardware-enforced DEP, DEP is automatically enabled only for operating system components. This is the default setting for client versions of Windows. DEP can be explicitly enabled for selected applications or the current process.
* OptOut (3)
  + DEP is automatically enabled for operating system components and all processes. This is the default setting for Windows Server versions. DEP can be explicitly disabled for selected applications or the current process. System compatibility fixes for DEP are in effect.

As mentioned in the “Validation information and the encrypted volume encryption key” section of this paper earlier, the integrity of Boot Configuration Data (BCD) settings is protected by full volume encryption (FVE).

### Process memory virtualization

The fourth level of protection is the process memory virtualization. Every process is given its own virtual memory space for operations. For each process, the Windows OS memory manager (residing in the kernel mode code) is responsible to map pages in virtual memory to pages in physical memory, so that physical memory addresses are not visible directly to the user mode code of the process. Within the user mode portion of the process virtual memory space, there are

* the user mode code for executing the process functional logics;
* specific data used by the process user mode code;
* user mode dynamic link libraries (DLLs) loaded directly or indirectly by the user mode code.

Due to this process memory virtualization, the virtual memory addresses of a process are not visible directly to another process. The process user mode code has free access to data residing in its own process virtual memory space, but not data residing in the process virtual memory space of another process. If the user mode code of Process A wishes to have access to a virtual memory address of Process B, then the user SID associated with the current execution thread of Process A or the user SID associated with Process A must have the necessary DACL permissions for accessing Process B. In addition, the MIC integrity level of Process A also must dominate the MIC integrity level of Process B. These process access permissions and MIC integrity level domination are enforced by the Windows OS process manager in the kernel mode. The DACL permission will be discussed further in the following section of this paper.

* “Meeting the “Access Control Polices Discretionary Access Control Policy Functional Requirements””.

The MIC integrity level domination will be discussed further in the following section of this paper.

* “Meeting the “Access Control Polices Mandatory Integrity Control Policy Functional Requirements””.

If data residing in the process virtual memory space of another process is deemed as security-relevant data, then it is protected from unauthorized access.

### Window terminal session isolation

The fifth level of protection is the window terminal session isolation. A window terminal session provides a boundary for all child processes, due to a successful remote or local interactive user logging on authentication, to reside in. All these child processes are assigned the association with the user SID of the logon user. The session itself is assigned a session ID which is unique within the local Windows OS machine, even though Session ID recycling also occurs. All Windows OS processes of the session are also assigned the session ID, which is used for subsequent session identification purpose. All processes of a session share the same windowing environment, which is maintained and managed by the session’s instance of the Windows OS window manager. The session’s instance of the Windows OS window manager runs in the kernel mode.

The Session ID assignment is managed and maintained by the Windows OS memory manager in the kernel mode. After a window terminal session is terminated, the Windows OS memory manger recycles the Session ID, which previously was assigned to the terminated window terminal session.

The Windows OS memory manager isolates the window manager instance of a session from the window manager instance of another session. For each process belonging to a session, the Windows OS memory manager maps the session’s instance of the Windows OS window manager into the session-specific location within the kernel mode portion of the process virtual memory space. Therefore, through the user mode kernel mode context switching, a process interacts with only the window manager instance of the session, where the process belongs to.

System service processes, which do not need to handle graphical user interaction, reside in a separate session, namely, Session 0. These system service processes are those managed by the Windows OS service control manager (services.exe).

Every instance of the window manager provides communication mechanisms such as window messages, hooks, and clipboard in the windowing environment that it maintains and manages. Due to the window terminal session isolation, processes are not free to use the communication mechanisms of the window manager instance to communicate with processes of a different session.

An established session can be disconnected and reconnected later on upon user re-authentication. The Windows OS local session manager (lsm.exe) and Windows OS remote window terminal service (termsrv.dll) together mediate the authorized access to a disconnected session and the authorized shadowing of a connected session. Both the Windows OS local session manager and remote window terminal service reside in Session 0. They execute in their own Windows OS processes.

On a Windows OS machine, there is only one local window terminal session that is connected to the local monitor and input devices such as keyboard, mouse, and/or others, so that a local interactive user interacts with the session. The Windows OS local session manager and remote window terminal service together maintain and manage the states of sessions created in the Windows OS machine. When the Windows OS local session manager disconnects the current local window terminal session from the local monitor and input devices, the window terminal session becomes disconnected. A disconnected session is not visible and it is not capable to receive input. After the previous local session is disconnected, another session is connected to the local monitor and input devices, for becoming the current local session.

In a Windows OS machine, multiple remote window terminal sessions are possible. The remote window terminal service controls the access to a remote window terminal session. A remote window terminal session is connected to a network stack, which delivers graphical elements of the session to and receives input elements from the authenticated remote interactive user.

Certain named resources can be bound to a specific session. When there is a request for an access to a named resource, the manager of the resource queries the source session of the request. If the source session of the request does not match the specific session, which was bound to the resource, then the access is denied.

If data residing in or resources associated with an established session are deemed as security-relevant data, then they are protected from unauthorized access.

### Secure desktop separation

The sixth level of protection is the secure desktop separation. Within a given window terminal session, there are a minimum of three desktops. They are the secure desktop, application desktop and screen saver desktop. These desktops are securable, named objects maintained by the session’s window manager, but their owner is the session’s window logon state maintaining service (aka winlogon.exe). The session’s window logon state maintaining service is a system process, which manages the transitions of states within the session. At any moment, there is only one current desktop. The current desktop is available for user interaction, which encompasses the acceptance of user input and the display of the desktop’s graphical elements. On the other hand, a non-current desktop is hidden and is not available for user interaction. As the owner of the desktops, the session’s window logon state maintaining service is responsible for instructing the session’s window manager to switch a specific desktop to become the current desktop, based on the current state condition(s).

Secure processes/applications such as user credential or explicit consent collecting applications live in the secure desktop to collect security sensitive data or explicit consent for action from the interactive user. Normal user applications live in the application desktop to interact with the interactive user. A screen saver application lives in the screen saver desktop so that normal user applications on the application desktop can be hidden effectively, as the viewing of and the interactions with applications of the application desktop are not available while the screen saver desktop is displayed.

Specifically, as the owner of the secure desktop, the session’s window logon state maintaining service limits the access to the secure desktop, so that a subject (other than a system process) is not permitted to target:

* the application processes created by the window logon state maintaining service for running on the secure desktop;
* the window elements of the created application processes,

for a window manager operation through a request to the window manager of the same window terminal session, where the subject resides in.

If data residing on the secure desktop or entered by the interactive user through an application living in the secure desktop is deemed as security-relevant data, then it is protected from unauthorized access.

### Isolation of user-entered credential through encryption

A level of protection provided on the top of the “secure desktop separation” level of protection is the isolation of user-entered credential through encryption.

The window logon user interface service (aka LogonUI.exe) is a system process living in the secure desktop. It collects the credential information (such as the user name with a password or the smart card PIN) corresponding to a credential provider installed in the local machine, from the interactive user. Upon receiving the credential information, LogonUI encrypts the credential information (with a short-live per-boot encryption key), so that only the Windows OS credential verification specific component(s) residing within the Windows OS authentication service needing the user credential in clear text for verification purpose would be able to conduct a successful decryption.

As a result, the user-entered credential information remains encrypted (and therefore is isolated), as it may have to travel to different parts of the Windows OS through the Windows OS inter-process communication mechanisms.

As user-entered credential information is deemed as security-relevant data, it is protected from unauthorized access.

### User interface privilege isolation within the same desktop

Another level of protection provided on the top of the “window terminal session isolation” level of protection is the user interface privilege isolation within the same desktop. As multiple user applications live in the application desktop to interact with the interactive user, they potentially belong to different MIC integrity levels. User interface privilege isolation is enforced by the window manager when the window manager handles and processes an application’s request to use a window manager’s communication mechanism to communicate with another application. The MIC integrity levels are discussed further in the following section of this paper.

* “Meeting the “Access Control Polices Mandatory Integrity Control Policy Functional Requirements””.

User interface privilege isolation blocks a lower integrity application from

* using the handle of a window element belonging to another application of a higher integrity level for specific operations;
* sending or posting certain window messages to a window element belonging to another application of a higher integrity level, unless the window messages are explicitly exempted by the message filter associated with the higher integrity application;
* using hooks to attach to or monitor another application of a higher integrity level;
* receiving raw keyboard or mouse data from the foreground message queue while an application of a higher integrity level is being attached to the message queue;
* attaching a message queue of an application of a higher integrity level to a message queue of another application of a higher integrity level.

User interface privilege isolation also blocks a higher integrity application from

* using a clip, sourced from an application of lower integrity level, from the clipboard.

If an application of a higher integrity level is deemed as security relevant, then it is protected from unauthorized access.

### Persistent data access mediation

The Windows OS Active Directory, Windows OS security audit store, Windows OS WMI data store, Windows OS registry, and Windows OS NTFS volume(s) are the primary storage locations where persistent data is stored. Access to these storage locations is mediated to prevent unauthorized access.

In the case of the Windows OS registry or Windows OS NTFS volume(s), their access mediation occurs in the kernel mode code of the requesting process. The user mode kernel mode context switching ensures that this access mediation is not by-passed.

In the case of the Windows OS Active Directory, Windows OS security audit store, Windows OS WMI data store, their access mediation occurs in the separate processes of their resources managers. The requesting process needs to use one of the inter-process communication mechanisms provided by the Windows OS to make a request to the resource manager processes. A request handler within a resource manager decides to grant the specific access request, based on the corresponding access policy, before returning either a negative (i.e. access denial) or a positive (i.e. access request having been carried out) response over the inter-process communication. If the resource manager process resides in the same Windows OS machine as the requesting process, then the process memory virtualization ensures that the access mediation is not by-passed. If the resource manager process resides in the Windows OS machine different from the requesting process, then the (remote) user authentication ensures that the access mediation is not by-passed.

If data stored in the Windows OS Active Directory, Windows OS security audit store, Windows OS WMI data store, Windows OS registry, and Windows OS NTFS volume(s) is deemed as security-relevant data, then it is protected from unauthorized access.

### Encrypting file system (EFS)

The final level of protection, for persistent user data in files, is encrypting file system, which is an integral part of the Windows OS native file system (NTFS) for accessing a local NTFS volume and of the Windows OS native [SMB](http://msdn2.microsoft.com/en-us/library/cc246231.aspx) remote file system redirector for accessing a remote NTFS volume across the network. As a subject is writing to or reading from a file residing in a local of remote NTFS volume, the file data is locally encrypted or decrypted, respectively. During encryption, the file encryption key is protected by a public key certificate associated with the file writing subject. If the file writing subject wishes to share the file with other subjects, then the file encryption key are additionally protected by public key certificates associated with those sharing subjects. During decryption, a subject must possess a private key, which is capable to recover the file encryption key from its protected form. If the subject cannot recover the file encryption key, then the clear text file data is denied to the subject. The encrypting file system is discussed further in the following section of this paper.

* “Meeting the “Import/Export of Data Exported Data Functional Requirements””.

If the user data files being encrypted are deemed as security-relevant data, then the encrypting file system relies on a valid file encryption key recovering private key to protect the encrypted files from unauthorized access.

### Long lived cryptographic key isolation service

In previous versions of Windows, clear text long lived cryptographic keys have to reside in the user mode Windows processes of applications as the applications need to perform specific cryptographic operations (such as decryption, hash signing for digital signature, key agreement, and key export) involving the cryptographic keys. Certain customers have been longing for a smart card like behavior in which clear text long lived cryptographic keys do not need to reside in the user mode Windows processes of applications while exercising the cryptographic services involving the cryptographic keys. This limitation is overcome in Windows Vista and Windows Server 2008 with the availability of the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx). The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) is hosted in the LSASS.exe Windows OS process (i.e. the same process that also hosts the Windows OS Authentication Service). The service provides cryptographic key process isolation to cryptographic public/private key pairs and associated cryptographic operations for opening, creation, deletion, import, export, encrypting, decrypting, hash signing, signature verification, and secret agreement.

If clear text long lived cryptographic keys are deemed as security-relevant data, then the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) provides the necessary process isolation so that user mode Windows OS processes of applications do not need to have a direct access to the clear text long lived cryptographic keys while requesting their associated cryptographic services and operations.

As it will be discussed in the “Key isolation service” section of this paper, the decision for the authorization to use long lived cryptographic keys in their associated cryptographic services and operations still relies on the persistent data access mediation in Windows OS NTFS volume(s) for the storage of the associated key files and the services of the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx). Therefore, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) is deemed as another level of protection, being implemented on top of the user authentication and the persistent data access mediation, under the “defense in depth” concept.

Consequently, this Commercial Grade OS Requirement Set “1.1.1.1” requirement is addressed.

## Addressing 1.1.1.2 “The OS shall ensure that all security policies are enforced before each security function is allowed to proceed”

For each level of protection provided under the “defense in depth” concept, we describe, in this section, the “domain” locations where the corresponding security policies are enforced when a request for access to the specific protected data is attempted.

### Full volume encryption

In the case of the full volume encryption protection, its security policy enforcement occurs in the Windows OS boot manager (bootmgr.efi on EFI firmware machines or bootmgr.exe on PC/AT firmware machines). After the user physically pushes the power button of the machine, on which the Windows OS resides, the Windows OS boot manager takes over the execution control of the machine. A full volume encryption (FVE) protected volume is marked “-FVE-FS-” in its boot sector.

The Windows OS boot manager implements a set of internal functions to handle the IO operations for a FVE-protected volume. Among the internal functions, the “Fve open” function and the “Fve read” function are critical for gaining authorized access to the data in the FVE-protected volume.

#### TPM unsealing during volume open

The “Fve open” function gets the PIN from the user or the startup key in a USB flash drive, which is inserted by the user to the machine. The user PIN or the USB startup key then is used to get the volume master key through the [Trusted Platform Module (TPM)](http://download.microsoft.com/download/5/D/6/5D6EAF2B-7DDF-476B-93DC-7CF0072878E6/secure-start_exec.doc) unsealing command ordinal [“TPM\_ORD\_Unseal”](http://msdn.microsoft.com/en-us/library/aa376205(VS.85).aspx).

If this TPM unsealing operation fails due to an incorrect PIN, then the user is displayed the “fve-bad-pin” message, which is the following.

“That PIN is incorrect. If you entered the PIN with the number keys, use the function keys F1 - F10 instead. Press ENTER to try another PIN. Press ESC for recovery.”

The TPM unsealing operation ensures that the selected boot components specified under the “[TPM platform validation profile](http://download.microsoft.com/download/a/f/7/af7777e5-7dcd-4800-8a0a-b18336565f5b/BitLockerFlow.doc)” are not changed before returning the volume master key to the Windows OS boot manager. The boot components covered by the “TPM platform validation profile” are:

* Core Root of Trust of Measurement (CRTM), BIOS, and Platform Extensions, which, by default, is checked for not having been changed during the TPM unsealing operation;
* Platform and Motherboard Configuration and Data, which, by default, is not checked for not having been changed during the TPM unsealing operation;
* Option ROM Code, which, by default, is checked for not having been changed during the TPM unsealing operation;
* Option ROM Configuration and Data, which, by default, is not checked for not having been changed during the TPM unsealing operation;
* Master Boot Record (MBR) Code, which, by default, is checked for not having been changed during the TPM unsealing operation;
* Master Boot Record (MBR) Partition Table, which, by default, is not checked for not having been changed during the TPM unsealing operation;
* State Transition and Wake Events, which, by default, is not checked for not having been changed during the TPM unsealing operation;
* Computer Manufacturer-Specific, which, by default, is not checked for not having been changed during the TPM unsealing operation;
* NTFS Boot Sector, which, by default, is checked for not having been changed during the TPM unsealing operation;
* NTFS Boot Block, which, by default, is checked for not having been changed during the TPM unsealing operation;
* Boot Manager, which, by default, is checked for not having been changed during the TPM unsealing operation;
* BitLocker™ Access Control[[1]](#footnote-2), which, by default, is checked for not having been changed during the TPM unsealing operation.

The TPM unsealing operation fails if any of the selected boot components specified under the “TPM platform validation profile” has been changed.

#### Validation information and the encrypted volume encryption key

The target volume stores the FVE\_INFORMATION data structure and the FVE\_VALIDATION data structure. The FVE\_VALIDATION data structure contains the IntegrityCheck FVE\_DATUM, which is used to check the integrity of the FVE\_INFORMATION data structure of the target volume.

The Windows OS boot manager checks the AESCCM integrity hash using the volume master key as follows. The IntegrityCheck FVE\_DATUM is decrypted with the volume master key using the AESCCM algorithm to recover the inner hash. This inner hash must be the same as the computed SHA256 hash of the FVE\_INFORMATION structure.

The FVE\_INFORMATION structure contains a number of FVE\_DATASET data structures. One FVE\_DATASET structure is the FVE\_DATUM\_VALIDATION\_INFO structure. Another is the encrypted volume encryption key. The encrypted volume encryption key is processed to recover the clear text volume encryption key. The volume encryption key should have been encrypted by the volume master key using the AESCCM algorithm. If the clear text volume encryption key cannot be recovered from the encrypted volume encryption key, then the volume would remain “locked”. If recoverable, the clear text volume encryption key is cached within the Windows OS boot manager in the temporary “KeyData” variable associated with the volume. This clear text volume encryption key is used, later by the Windows OS boot manager, during the volume read function (namely, the “Fve read” function).

The FVE\_DATUM\_VALIDATION\_INFO structure contains

* the SHA256 hash of a boot application that is allowed to have access to the target volume;
* a SHA256 hash of the specific element of a configuration setting for the associated boot application, where the setting element is not marked as FVE\_VALIDATION\_BCD\_POLICY\_IGNORE under the corresponding “common ignore setting element” policy, “loader ignore setting element” policy, “resume ignore setting element” policy, and “memory test ignore setting element” policy.

Before the boot manager transfers the execution control to one of the following boot applications:

* winload.exe (on PC/AT firmware machines) or winload.efi (on EFI firmware machines), which is the application to load the Windows OS, namely ntoskrnl.exe;
* winresume.exe (on PC/AT firmware machines) or winresume.efi (on EFI firmware machines), which is the application for resuming the Windows OS from the hibernation file “hiberfil.sys”;
* memtest.exe (on PC/AT firmware machines) or memtest.efi (on EFI firmware machines), which is a memory testing application,

the application and its configuration setting elements are checked in the boot manager’s internal “Fve Datum Validate Code” function against the FVE\_DATUM\_VALIDATION\_INFO from the volume, where the application is associated with. If the boot manager fails to find an entry in the FVE\_DATUM\_VALIDATION\_INFO for the application or the application’s setting elements, then the user is displayed the following “fve-bad-code-id” message or the “fve-bad-code-options” message. These display messages include the identifier of the offending application and the code representation of the offending [Boot Configuration Data (BCD) setting element](http://msdn.microsoft.com/en-us/library/aa964229(VS.85).aspx).

* The “fve-bad-code-id” message:
  + “This boot application has changed since BitLocker was enabled. You must supply a BitLocker recovery password to start this system. Confirm that the changes to this boot application are trusted. If the changes are trusted then disable and re-enable BitLocker. This will reset BitLocker to use the new boot application information. Otherwise re-install the previous version of this boot application.”
* The “fve-bad-code-options” message:
  + “The Boot Configuration Data (BCD) settings for the following boot application have changed since BitLocker was enabled. You must supply a BitLocker recovery password to start this system. Confirm that the changes to the BCD settings are trusted. If the changes are trusted then disable and re-enable BitLocker. This will reset BitLocker to use the new BCD settings. Otherwise restore the original BCD settings.”

#### Decryption during reading of the opened volume

To read IO data from the opened (i.e. unlocked) volume, the “Fve read” function does the decryption in its internal “Read and Decrypt” function. Recall that the clear text volume encryption key has been cached within the Windows OS boot manager in the temporary “KeyData” variable associated with the volume. Therefore, the key is available for use in the internal “Read and Decrypt” function. If the corresponding decryption fails, only zeros are returned.

The detail about the various encryption/decryption algorithms can be found in the following Microsoft publication: <http://download.microsoft.com/download/0/2/3/0238acaf-d3bf-4a6d-b3d6-0a0be4bbb36e/BitLockerCipher200608.pdf>.

#### Passing the volume master key from the boot manager to the fvevol.sys filter driver

After obtaining the clear text volume encryption key using the clear text volume master key, the boot manager caches the clear text volume master key (marking it FVE\_ROLE\_VMK\_CACHE) into its temporary “KeyRing” variable. The value in this “KeyRing” variable is mapped into the physical memory at the “keyring” physical address. The actual “keyring” physical address is passed from the boot manager to the fvevol.sys filter driver via the registry key value named “HKLM\System\CurrentControlSet\Control\SystemStartOptions” with the “FVEBOOT=” identifier.

The fvevol.sys filter driver reads the content (which is a FVE\_KEYRING structure) of the “keyring” physical address and to store the content in its internal global variable “FveKeyring” for later use. The fvevol.sys filter driver extracts the clear text volume master key from the FVE\_KEYRING structure content.

Using the clear text volume master key, the fvevol.sys filter driver gets the clear text volume encryption key from the encrypted format residing in the FVE\_INFORMATION structure, which is read directly from the volume. The fvevol.sys filter driver caches the clear text volume encryption key in its internal “DevExt->FvekData” variable associated with the volume.

The Memory Overwrite Request (MOR) bit is available from the firmware (e.g. BIOS and EFI or PC/AT) that supports TPM. It is a special non-volatile bit used to tell the BIOS that the system memory should be erased early in the boot process during the next restart. The boot manager sets this MOR special bit before mapping the content of the internal “KeyRing” variable into the physical memory at the “keyring” physical address. The fvevol.sys filter driver also sets this bit after reading the FVE\_KEYRING content from the “keyring” physical address.

#### Volume encryption and decryption occurring in fvevol.sys filter driver

The fvevol.sys filter driver is a filter driver layering on the top of the volume manager (volmgr.sys), as shown in the following entry of its manifest. Therefore, it resides below any installed file systems of the Windows OS.

<categoryMembership>

<id name="Microsoft.Windows.Categories" version="1.0.0.0" publicKeyToken="365143bb27e7ac8b" typeName="Storage Volume"/>

<categoryInstance subcategory="LowerFilters">

<filter name="fvevol"

position="last"

/>

</categoryInstance>

</categoryMembership>

The fvevol.sys filter driver starts to filter data for the read IO from and data for the write IO to a volume when it is informed by the Windows OS plug and play manager that a new volume has been mounted to the machine. Obviously, the system volume (which contains the running Windows OS) would be the first volume that the Windows OS plug and player manager informs the fvevol.sys filter driver about the arrival. The fvevol.sys filter driver conducts the initialization for the volume with the volume’s corresponding [DEVICE\_EXTENSION](http://msdn.microsoft.com/en-us/library/ms794734.aspx) DevExt structure. The installation of the clear text volume encryption key into the fvevol.sys filter driver’s internal “DevExt->FvekData” variable takes place during the initialization as mentioned in the above.

Once the target volume has been unlocked through the initialization, subsequent writing to the system volume needs to be encrypted and subsequent reading from the system volume needs to be decrypted. The writing occurs in the fvevol.sys filter driver’s “Write Encrypt” function and the reading occurs in the fvevol.sys filter driver’s “Read Decrypt” function. Both encryption and decryption use the clear text volume encryption key stored in the “DevExt->FvekData” variable.

#### Summary of Full Volume Encryption security policy enforcement

The above has explained the enforcement of the security policies of the full volume encryption protection. In summary, the boot manager obtains the clear text volume master key under the specific conditions. The fvevol.sys filter driver ensures that the data written to the system volume is encrypted with the volume encryption key.

### User authentication

In the case of the user authentication, its security policy enforcement occurs in the Windows OS Authentication Service (aka lsass.exe). As mentioned in the above overview of the user authentication, every Windows OS process or thread created by the Windows OS process manager is assigned the association with a user SID. This user SID association is captured in a securable Windows OS object called “[access token](http://msdn.microsoft.com/en-us/library/aa374909.aspx)”. An access token is basically the “security context” associated with a process or a thread. While the Windows OS security reference monitor (SRM) in the kernel mode has the responsibility to manage and maintain access tokens, it is the Windows OS Authentication Service, which is responsible to request the Windows OS SRM to create an access token after it successfully authenticates a (local or remote) user. The request to create an access token is a privileged operation. The requester must possess the [SeCreateTokenPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK). By default, only the Windows OS Authentication Service possesses the [SeCreateTokenPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK).

#### Authentication Protocols

Before the Windows OS Authentication Service requests to create an access token, it must complete the authentication process for a user based on one of the supported authentication protocols. For the Windows OS, the supported authentication protocols are:

* [Kerberos](http://msdn2.microsoft.com/en-us/library/aa378747(VS.85).aspx), which is implemented in Kerberos.dll on an individual machine and in kdcsvc.dll on a server (such as a Windows OS domain controller) designated as a Kerberos key distribution center (KDC);
* [NTLM](http://msdn2.microsoft.com/en-us/library/aa378749(VS.85).aspx), which is implemented in msv1\_0.dll on an individual machine;
* [Netlogon](http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/%5BMS-NRPC%5D.pdf), which is implemented in netlogon.dll on a domain joined machine
  + It is listed here for completeness only, as it does not handle user authentication directly, but it is responsible for maintaining the secure channels among machines joined to a Windows OS domain;
* [CredSSP](http://msdn2.microsoft.com/en-us/library/bb931352(VS.85).aspx), which is implemented in TSpkg.dll on an individual machine;
* [RFC 2617 Digest Access Protocol](http://msdn2.microsoft.com/en-us/library/aa378745(VS.85).aspx), which is implemented in wdigest.dll on an individual machine;
* [RFC 2246 Transport Layer Security (TLS)](http://msdn2.microsoft.com/en-us/library/aa380123(VS.85).aspx), which is implemented in Schannel.dll on an individual machine;
* Microsoft Clear Text Password Security Provider, which is implemented in PwdSSP.dll on a Windows OS domain controller machine
  + It is listed here for completeness only, as the reason of its existence is to support LDAP simple bind for the LDAP server residing on a Windows OS domain controller; and, in any case, it uses other available security providers to complete its user authentication tasks;
* ADFS security provider, which is implemented in IfsAp.dll on an individual server machine of the ADFS Web Agent Authentication Service, where the Kerberos.dll does not support [KERB\_S4U\_LOGON](http://msdn.microsoft.com/en-us/library/aa378128(VS.85).aspx)
  + It is listed here for completeness only, as the Kerberos.dll shipping in Windows Server 2008 supports [KERB\_S4U\_LOGON](http://msdn.microsoft.com/en-us/library/aa378128(VS.85).aspx).

The implementations of all the above are security providers (SPs) or security support providers (SSPs) according to the SSPI model explained in [MSDN](http://msdn2.microsoft.com/en-us/library/aa380497(VS.85).aspx). They are loaded into the Windows OS process address space of the Windows OS Authentication Service. Therefore, they are parts of the Windows OS Authentication Service.

#### User credential materials

To process a user authentication, certain user credential materials must be input into one of the above security providers. The following two sets of functions, defined in the [SECPKG\_FUNCTION\_TABLE](http://msdn.microsoft.com/en-us/library/aa380114.aspx) structure, are exported by a security provider, through the Windows OS Authentication Service, for accepting user credential materials:

* [LogonUser()](http://msdn.microsoft.com/en-us/library/aa378184(VS.85).aspx), or [LogonUserEx()](http://msdn.microsoft.com/en-us/library/aa378189(VS.85).aspx), or [LogonUserEx2()](http://msdn.microsoft.com/en-us/library/aa378257(VS.85).aspx);
* [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx).

These two sets of functions would produce a handle to a primary or an impersonation access token for associating with a Windows OS process or a thread.

In the case of [LogonUserXXX()](http://msdn.microsoft.com/en-us/library/aa378257(VS.85).aspx), the typical user credential materials are “a user name with a password”, or a “smart card PIN”, or certain security protocol specific input messages such as [KERB\_S4U\_LOGON](http://msdn.microsoft.com/en-us/library/aa378128(VS.85).aspx), or [MSV1\_0\_LM20\_LOGON](http://msdn.microsoft.com/en-us/library/aa378762(VS.85).aspx). To use the [LogonUserXXX()](http://msdn.microsoft.com/en-us/library/aa378257(VS.85).aspx) functions for acquiring an access token, a Windows OS process must register itself as a logon process through the use of [LsaRegisterLogonProcess(](http://msdn.microsoft.com/en-us/library/aa378318.aspx)), which requires the Windows OS process to possess the [SeTcbPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK). Another reason why the [SeTcbPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) is required is that the user credential materials are security sensitive materials and they could be subject to the “dictionary” attack through the calling of [LogonUserXXX()](http://msdn.microsoft.com/en-us/library/aa378257(VS.85).aspx). The Windows OS Authentication Service assumes that a registered Windows OS logon process has a reliable mechanism to receive the necessary user credential materials from the user. For example, the Session X Window Logon State Maintaining Service (winlogon.exe) is a registered Windows OS logon process. It relies on a “trusted path” mechanism to receive the necessary user credential materials from the user.

In the case of [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx), the typical user credential materials are cryptographic responses due to some initial challenges according to the specific security protocols. Examples of these cryptographic responses are Kerberos Application Requests ([KRB\_AP\_REQ](http://msdn.microsoft.com/en-us/library/cc212328.aspx)), or NTLM response messages, or RFC 2617 Digest Access Protocol response messages, or TLS Handshake Protocol client messages. These cryptography-protected responses are expected to be safe from cryptographic attacks (including the “man in the middle” attack). Therefore, unlike the case of [LogonUserXXX()](http://msdn.microsoft.com/en-us/library/aa378257(VS.85).aspx), a caller of [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx) is not required to have the [SeTcbPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK). Typical callers of [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx) are Windows OS processes which are Windows OS services running in the security context of local system, local service, or network service. Furthermore, calling [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx) from the kernel mode is also supported. These Windows OS services have remote network interfaces to serve their specific network clients. These remote network interfaces typically require their specific security levels of user authentication. The role of [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx) is to accomplish the user authentication requirements associated with the remote network interfaces. The access tokens produced by [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx) are impersonation tokens. If a caller of [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx) wishes to use the impersonation token to impersonate the user associated with the impersonation token, then the caller must have the [SeImpersonatePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK). Otherwise, the impersonation token is only good for the user identification purpose. By default, local system, local service, and network service have the [SeImpersonatePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK).

#### Registered logon processes

Within the Windows OS, the following are Windows OS processes that register themselves as a logon process:

* ServerForNFS, which belongs to nfssvr.sys (for providing the UNIX NFS service as parts of the services for UNIX);
* ClientForNFS, which belongs to nfsrdr.sys (for providing the UNIX NFS redirection service as parts of the services for UNIX);
* Secondary Logon Service, which belongs to seclogon.dll (for an interactive user to start an application using the security context of another user account);
* ADFS Web Agent Authentication Service, which belongs to ifssvc.dll (for supporting the web client user authentication based on the use of [ADFS SAML security token](http://msdn2.microsoft.com/en-us/library/bb897402.aspx));
* Session X Window Logon State Maintaining Service, which belongs to Session X winlogon.exe (for an initial interactive user logon locally or remotely);
* Consent UI application, which belongs to consent.exe (for the Administrative Privileged Application Launching Service to validate the interactive user entered logon credentials for an authorized user account that has the permission to conduct a Windows OS administrative action);
* Kernel mode http handler, which belongs to http.sys (for supporting the authentication of a http client who selects the http basic authentication);
* Internet Authentication Service (IAS), which belongs to iassam.dll and raschap.dll being loaded in ias.dll (for supporting the user authentication of remote access through the Windows OS’s implementation of a Remote Authentication Dial-in User Service (RADIUS) server);
* IPSec Internet Key Exchange (IKE), which belongs to ikeext.dll (for supporting the authentication of peer machines when establishing IPSEC protected connections with peer machines)
  + A Windows OS IPSec component is hooked to the TCP/IP network stack to provide network access protection as well as protocol filtering.

#### Windows OS services needing user authentication when handling network requests

On the other hand, the following are Windows OS processes that use [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx) (i.e. [AcceptSecurityContext()](http://msdn.microsoft.com/en-us/library/aa374703(VS.85).aspx)):

* Microsoft Virtual System Migration Service over TCP port 6600, which belongs to vmms.exe (for supporting the machine-authentication in the protocol for receiving a migrated “Virtual Machine” from another host);
* WS-Management over HTTP, which belongs to WsmSvc.dll (for supporting the user authentication in the WS-Management protocol);
* Microsoft Failover Cluster Service, which belongs to clussvc.exe (for supporting the authentication of a machine belonging to a machine cluster);
* Server drivers, which belongs to srv2.sys, srvnet.sys, and srv.sys (for supporting the user authentication in remote file system services over SMB and SMB V2);
* Boot Information Negotiation Layer of the Remote Installation service, which belongs to binlsvc.dll;
* Any RPC server listener using the RPC run time (rpcrt4.dll) functions such as [RpcServerRegisterIfEx()](http://msdn.microsoft.com/en-us/library/aa378441(VS.85).aspx) or similar to register a RPC interface spec with a security callback function;
* Any DCOM server listener using the DCOM run time (ole32.dll) functions such as [CoInitializeSecurity()](http://msdn.microsoft.com/en-us/library/ms693736(VS.85).aspx) or similar to define the client authentication and security requirements;
* Any .NET applications, including web services, that use the [System.Web.Security](http://msdn.microsoft.com/en-us/library/system.web.security.aspx) classes;
* DNS server, which belongs to dns.exe (for supporting peer authentication when answering DNS record update PDUs);
* LDAP server, which belongs to ntdsa.dll and ntdsai.dll (for supporting user authentication in the LDAP protocol);
* IIS web server, which uses the built-in authsspi.dll and authmd5.dll HTTP modules (for supporting web client authentication over [Kerberos](http://msdn2.microsoft.com/en-us/library/aa378747(VS.85).aspx), [NTLM](http://msdn2.microsoft.com/en-us/library/aa378749(VS.85).aspx), or [RFC 2617 Digest Access Protocol](http://msdn2.microsoft.com/en-us/library/aa378745(VS.85).aspx));
* Kerberos based FTP Server, which belongs to ftpsvc2.dll;
* Microsoft Message Queue (MSMQ), which belongs to mqsec.dll (for supporting user authentication in the MSMQ protocol);
* Windows Media Services, which belongs to WMSServer.dll (for supporting authentication of a user who requests multi-media contents from the Windows Media Services);
* Kernel mode http handler, which belongs to http.sys (for supporting the authentication of a http client who selects TLS, Kerberos, or NTLM);
* IPSec Internet Key Exchange (IKE), which belongs to ikeext.dll (for supporting the authentication of peer machines when establishing IPSEC protected connections with peer machines)
  + A Windows OS IPSec component is hooked to the TCP/IP network stack to provide network access protection as well as protocol filtering;
* Peer-to-Peer Networking Services, which belongs to p2psvc.dll;
* Internet Authentication Service (IAS), which belongs to rastls.dll being loaded in ias.dll (for supporting the user authentication of remote access through the Windows OS’s implementation of a Remote Authentication Dial-in User Service (RADIUS) server);
* Remote Terminal Session Service Security Driver, which belongs to tssecsrv.sys (for support the user authentication based on [RFC 2246 Transport Layer Security (TLS)](http://msdn2.microsoft.com/en-us/library/aa380123(VS.85).aspx) or [CredSSP](http://msdn2.microsoft.com/en-us/library/bb931352(VS.85).aspx) associated with the RDP protocol).

#### Summary of User Authentication security policy enforcement

As described earlier, the Windows OS Authentication Service is responsible for creating a primary or an impersonation token. On the other hand, a registered logon process or a Windows OS service, which has a remote network interface, receives the specific user credential materials from its client. The logon process or the Windows OS service must pass the received user credential materials from its client to the corresponding security providers of the Windows OS Authentication Service for validation, in order to obtain an access token to represent the client. Otherwise, the logon process or the Windows OS service is not able to have a Windows OS process or a Windows OS thread to run in the security context of the client. As a result, security relevant actions of the client do not occur on the Windows OS.

#### Session ID assignment to an access token

When the Windows OS Authentication Service requests the Windows OS security reference monitor (SRM) to create an access token, it needs to assign a Session ID to the access token. If the registered logon process or the Windows OS service requesting the token creation belongs to Session X, then Session X is the Session ID assigned to the access token. For example, if Session X Window Logon State Maintaining Service requests the Windows OS Authentication Service for a primary token to represent the interactive logon user, then Session X is assigned as the Session ID of the primary token.

### User mode kernel mode context switching

In the case of the user mode kernel mode context switching, its security policy enforcement occurs in the specific kernel mode interfaces of the Windows OS.

As described earlier, when the user mode code of a Windows OS process or thread attempts to gain access to kernel mode resources, it is necessary to enter specific kernel mode interfaces of the Windows OS, which conduct the user mode kernel mode context switching. After the context switching, the corresponding kernel mode code takes over the execution control. When the kernel mode code finishes its executions, it gives up its execution control so that the original context, which was maintained just before the context switching, is restored.

The specific kernel mode interfaces of the Windows OS are implemented in two system tables. The first is KiServiceTable. The second is W32pServiceTable. Both are installed under a fixed physical address.

The KiServiceTable defines interfaces provided by the Windows OS kernel (namely ntoskrnl.exe). They are local interfaces to the following Windows OS kernel components:

* the Windows OS configuration manager for the local registry (config.lib);
* the Windows OS kernel mode debug manager (dbgk.lib);
* the Windows OS event tracing manager (etw.lib);
* the Windows OS executive manager for computer management (ex.lib);
* the Windows OS IO manager (io.lib);
* the Windows OS kernel debugger (kd.lib);
* the Windows OS asynchronous local (inter) process communication manager (alpc.lib);
* the Windows OS memory manager (mm.lib);
* the Windows OS object manager (ob.lib);
* the Windows OS power manager (po.lib);
* the Windows OS process manager (ps.lib);
* the Windows OS security reference monitor (se.lib);
* the Windows OS transaction manager (tm.lib);
* the Windows OS virtual DOS machine (vdm.lib).

The W32pServiceTable defines interfaces provided by the Windows OS window manager (namely win32k.sys). They are local interfaces to the Windows OS window manager.

As a result, security relevant kernel mode actions of a user mode requester do not occur until the requester enters through one of the interfaces of KiServiceTable or W32pServiceTable.

### Process memory virtualization

In the case of the process memory virtualization, its security policy enforcement occurs in the Windows OS memory manager.

When the Windows OS process manager handles an authorized request to create a new Windows OS process, it turns to the Windows OS memory manager to create the virtual address space for the new process in the Windows OS memory manager’s “Create Process Address Space” function. Specifically, the Windows OS memory manager’s “Create Process Address Space” function provides the top level page directory page (i.e. the directory table base) of the newly created virtual address space for the new process. As a result, when the user mode code of the newly created process has the execution control, the virtual address space for the process has already been initialized for the user mode code to consume. Therefore, the security policy for the process memory virtualization is accomplished.

### Window terminal session isolation

In the case of the window terminal session isolation, its security policy enforcements occur in a number of locations, as explained below.

#### Session specific window manager isolation

The session specific window manager isolation is enforced by the Windows OS memory manager. The Windows OS memory manager isolates the window manager instance of a session from the window manager instance of another session. As the Windows OS memory manager creates the virtual address space for a new process in its “Create Process Address Space” function, it also adds the virtual address region, containing the Windows OS window manager instance of the session, being targeted for the new process, into the new process’s virtual address space. Therefore, when the user mode code of the newly created process has the execution control, only interfaces to the Windows OS window manager instance of the session being targeted for the new process are available to the user mode code.

#### Special privilege required when requesting to create a new process belonging to another session

When the Windows OS process manager handles a request of a new process creation, it allows the requester to specify a target access token for associating with the new process. If the Session ID of the requester and the session ID of the target access token are different, then the Windows OS process manager requires the requester to possess the [SeAssignPrimaryTokenPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK). By default, only the local system, local service and network service are given the [SeAssignPrimaryTokenPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK).

Recall that the Windows OS Authentication Service assigns the Session ID of a new access token being created based on the Session ID of the requesting logon process or Windows OS service. Therefore, if the requesting logon process or Windows OS service belongs to Session X, then the newly created access token is assigned Session X also. As a result, any Windows OS process created for associating with the newly created access token belongs to Session X also. Hence, processes belonging to a specific session are typically child processes created by the interactive user who has logged on the specific session. Consequently, Session ID could be used as an identification mechanism when limiting certain system resources to only the processes belonging to a specific session.

#### Reconnecting or shadowing a session

The Windows OS remote window terminal service (termsrv.dll), together with the assistance of the Windows OS local session manager (lsm.exe), mediates the authorized access to a disconnected window terminal session and the authorized shadowing of a connected window terminal session. Therefore, the graphical user interfaces of a window terminal session are accessible to authorized interactive users only.

#### Summary of Window terminal session isolation policy enforcement

The policy enforcement is summarized as follows.

* The window terminal session specific window manager isolation is enforced by the Windows OS memory manager.
* Creation of a Windows OS process belonging to another window terminal session different from the window terminal session of the requester is a privileged operation; and it is enforced by the Windows OS process manager.
* The authorized graphical user interface level of access to a local or remote window terminal session is mediated by the Windows OS remote window terminal service.

### Secure desktop separation

Within a window terminal session, the session’s window logon state maintaining service is the owner of the secure desktop of the window terminal session. The session’s window logon state maintaining service enforces the security policy that a subject (other than a system process) is not permitted to target:

* the application processes created by the window logon state maintaining service for running on the secure desktop;
* the window elements of the created application processes,

for a window manager operation through a request to the window manager of the same window terminal session where the subject resides in.

Furthermore, the session’s window logon state maintaining service switches to the session’s secure desktop for graphical user interaction whenever the local interactive user presses the “trusted path” hotkey on the keyboard, which has been registered with the session’s window manager. The “trusted path” hotkey registration occurs when the session’s window logon state maintaining service requests the session’s window manager to create the “Winsta0” windowstation to house the secure desktop, during its initialization. Because the hotkey registration occurs during the initialization of the session’s window logon state maintaining service, no one else belonging to the session could have made the same hotkey registration.

### Isolation of user-entered credential through encryption

Recall that the window logon user interface service (aka LogonUI.exe) is a system process living in the secure desktop. It collects the credential information (such as the user name with a password or the smart card PIN) corresponding to a credential provider installed on the local machine, from the interactive user. Upon receiving the credential information, the window logon user interface service encrypts the credential information (with a short-live per-boot encryption key, maintained by the Windows OS security driver, ksecdd.sys), so that only the Windows OS credential verification specific component(s) residing within the Windows OS authentication service needing the user credential in clear text for verification purpose would be able to conduct a successful decryption.

Therefore, the window logon user interface service enforces the security policy of credential information encryption upon receiving the credential information from the interactive user.

### User interface privilege isolation within the same desktop

The security policies for user interface privilege isolation are enforced by the window manager when the window manager handles and processes an application’s request to use a window manager’s communication mechanism to communicate with another application. These policies are stated as follows.

User interface privilege isolation blocks a lower integrity application from

* using the handle of a window element belonging to another application of a higher integrity level for specific operations;
* sending or posting certain window messages to a window element belonging to another application of a higher integrity level, unless the window messages are explicitly exempted by the message filter associated with the higher integrity application;
* using hooks to attach to or monitor another application of a higher integrity level;
* receiving raw keyboard or mouse data from the foreground message queue while an application of a higher integrity level is being attached to the message queue;
* attaching a message queue of an application of a higher integrity level to a message queue of another application of a higher integrity level.

User interface privilege isolation also blocks a higher integrity application from

* using a clip, sourced from an application of lower integrity level, from the clipboard.

### Persistent data access mediation

Recall that access to the Windows OS Active Directory, Windows OS security audit store, Windows OS WMI data store, Windows OS registry, and Windows OS NTFS volume storage locations is mediated to prevent unauthorized access.

In the case of the Windows OS registry or Windows OS NTFS volume, their access mediation occurs in the kernel mode code of the requesting process due to the kernel mode interfaces to the Windows OS configuration manager or the Windows OS IO manager respectively. The user mode kernel mode context switching ensures that this access mediation policy is not by-passed. As the access token associated with the requesting process or thread is available, the kernel mode interfaces use the process or thread access token in the access policy enforcement decision to decide the granting of the specific access request to the requesting process or thread.

In the case of the Windows OS Active Directory, Windows OS security audit store, and Windows OS WMI data store, their access mediation occurs in the separate processes of their resources managers. The requesting process needs to use one of the inter-process communication mechanisms provided by the Windows OS to make a request to the resource manager processes.

In the case of the Windows OS Active Directory, it is the LDAP server (belonging to ntdsa.dll and ntdsai.dll). Its inter-process communication mechanism is LDAP. As mentioned earlier, it uses the [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx) to accomplish the user authentication for obtaining an access token to represent the user.

In the case of the Windows OS security audit store, it is implemented in wevtsvc.dll. Its inter-process communication mechanism is [RPC](http://msdn2.microsoft.com/en-us/library/aa385462(VS.85).aspx). Therefore, the Windows OS security audit store is a RPC server listener. As mentioned earlier, it uses the [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx) to accomplish the user authentication for obtaining an access token to represent the user.

In the case of the Windows OS WMI data store, it is implemented in WMIsvc.dll. The Windows OS WMI data store registers the class ID [CLSID\_WbemLevel1Login](http://msdn2.microsoft.com/en-us/library/cc250755.aspx). Its inter-process communication mechanism is DCOM. Therefore, the Windows OS WMI data store is a DCOM server listener. As mentioned earlier, it uses the [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx) to accomplish the user authentication for obtaining an access token to represent the user.

In the case of the Windows OS Active Directory, Windows OS security audit store, and Windows OS WMI data store, there is a request handler within each of them. The request handler uses the access token, obtained from [AcceptLsaModeContext()](http://msdn.microsoft.com/en-us/library/aa380154(VS.85).aspx), in the access policy enforcement decision to decide the granting of the specific access request to the requester user (represented by the access token).

If the request handler resides in the same Windows OS machine as the requesting process, then the process memory virtualization ensures that the access policy enforcement is not by-passed. If the request handler resides in the Windows OS machine different from the requesting process, then the (remote) user authentication ensures that the access policy enforcement is not by-passed.

### Encrypting file system (EFS)

The encryption/decryption of EFS-protected files occurs in two locations. One is the NTFS file system (ntfs.sys). The other is the [SMB](http://msdn2.microsoft.com/en-us/library/cc246231.aspx) remote file system redirector (mrxsmb20.sys). Both reside in the kernel mode of the Windows OS. A subject needs to use the kernel mode interfaces to the Windows OS IO manager in order to gain access to their services. The NTFS file system supports the access of files residing locally. The [SMB](http://msdn2.microsoft.com/en-us/library/cc246231.aspx) remote file system redirector supports the access to files residing remotely across the network. In the SMB redirector case, file encryption and decryption occurs locally. Therefore, the redirector sends encrypted file data over SMB to a remote server for storage, and it receives encrypted file data from a remote server over SMB. The remote server processes the encrypted file data as raw file data only for storage.

Both the NTFS file system and the SMB redirector need to obtain a file encryption key when a subject requests to open an EFS-protected file through the kernel mode interfaces to the Windows OS IO manager for reading or writing. The file encryption key is obtained from the local EFS key management unit residing within the Windows OS authentication service process (lsass.exe). The communication to the local EFS key management unit is handled by the Windows OS asynchronous local (inter) process communication manager. The communication is limited to kernel mode originator only. The local EFS key management unit either

* generates a fresh file encryption key and then protects it with the subject’s cryptographic public/private key pair;
* recovers the file encryption key from its protected format using the subject’s cryptographic private key.

If the file encryption key cannot be recovered, then the subject is denied the access to the file which the subject attempts to open. The subject’s cryptographic private key either

* resides locally and is protected by the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) residing within the Windows OS authentication service process also;
* resides within a smart card and the local EFS key management unit launches the application efsui.exe on the subject’s window terminal session for collecting the smart card PIN from the subject.

If the “Enable Secure Credential Prompting” policy is enabled via the following registry key, then the UI dialog for collecting the smart card PIN appears on the secure desktop after the interactive user has been informed to press the “trusted path” hotkey.

* HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\CredUI\EnableSecureCredentialPrompting

Having obtaining the file encryption key, the NTFS file system or the SMB redirector is able to perform data decryption using the file encryption key, when conducting the read file operation for the subject. The NTFS file system or the SMB redirector is also able to perform data encryption using the file encryption key, when conducting the write file operation for the subject.

As a result, the combination of the NTFS file system, the SMB redirector, and the local EFS key management unit enforces the security policy that the file encryption key must be recovered from the subject’s cryptographic private key for opening an EFS protected file.

### Long lived cryptographic key isolation service

Recall that the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) is hosted in the LSASS.exe Windows OS process (i.e. the same process that also hosts the Windows OS Authentication Service). It relies on the persistent data access mediation in Windows OS NTFS volume(s) for the storage of the associated key files and the services of the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx). The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) enforces the security policy of long lived cryptographic key process isolation. When requesting cryptographic operations for opening, creation, deletion, import, export, encrypting, decrypting, hash signing, signature verification, and secret agreement associated with a long lived cryptographic public/private key pair, the authorization decision of the request occurs in the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx). In addition, the clear text of the long lived cryptographic private key only needs to reside inside the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx).

Consequently, this Commercial Grade OS Requirement Set “1.1.1.2” requirement is met.

## Addressing 1.1.1.3 “The OS shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects”

Having described the “domain” locations where the corresponding security policies are enforced, we explain that these “domain” locations are separated from interference and tampering by untrusted subjects.

### Full volume encryption

In the case of the full volume encryption (FVE) protection, the security domain location is the Windows OS boot manager (bootmgr.efi or bootmgr.exe). An untrusted user, at best, only has physical access to the FVE protected volume. The encryption due to the FVE protection prevents the untrusted user’s interference and tampering to the Windows OS and its associated data residing in the FVE protected volume.

For encrypting the data as the data is written to the FVE protected volume, the security domain is the Windows OS fvevol.sys filter driver, residing in the kernel mode. Without possessing additional privileges of an administrator to enter specific kernel mode interfaces successfully, an untrusted subject from the user mode is prevented from interfering and tampering the operations of the Windows OS fvevol.sys filter driver.

When the Windows OS is running after a successful reboot, interfaces to the underlying [Trusted Platform Module (TPM)](http://download.microsoft.com/download/5/D/6/5D6EAF2B-7DDF-476B-93DC-7CF0072878E6/secure-start_exec.doc) are visible to a local subject through the Windows OS [TPM Base Services (TBS)](http://msdn.microsoft.com/en-us/library/aa446792(VS.85).aspx) (tbssvc.dll). The Windows OS [TPM Base Services (TBS)](http://msdn.microsoft.com/en-us/library/aa446792(VS.85).aspx) mediates TPM Command Ordinals being sent to the TPM, as the Windows OS [TPM Base Services (TBS)](http://msdn.microsoft.com/en-us/library/aa446792(VS.85).aspx) is a Windows OS service that accepts TPM Ordinal commands over its local [RPC interface](http://msdn.microsoft.com/en-us/library/aa446799(VS.85).aspx). However, the Windows OS [TPM Base Services (TBS)](http://msdn.microsoft.com/en-us/library/aa446792(VS.85).aspx) always blocks certain TPM Ordinal commands for reaching the TPM from the user mode. The default list of the [blocked TPM Command Ordinals](http://msdn.microsoft.com/en-us/library/aa376205(VS.85).aspx) is stored in the following registry key as a list of registry values.

* “HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Tpm\BlockedCommands\List”.

After a submitted TPM Command Ordinal is blocked, the Windows OS [TPM Base Services (TBS)](http://msdn.microsoft.com/en-us/library/aa446792(VS.85).aspx) generates the [Event ID 4671](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_TBS\_BLOCKED\_ORDINAL\_value) “An application attempted to access a blocked ordinal through the TBS” audit record in the hard audit store to indicate the TPM Command Ordinal being blocked and the user account identifiers of the subject attempting the blocked TPM Command Ordinal submission.

### User authentication

In the case of the user authentication, the security domain is the Windows OS Authentication Service (lsass.exe). If the user authentication of a subject should fail, then the subject’s interaction is stopped at either a registered logon process or a Windows OS service having remote network interfaces, which is accepting user credential materials. Consequently, there would not be a process or a thread, running in the security context of the subject attempting the user authentication, to launch any interference and tampering actions.

### User mode kernel mode context switching

In the case of the user mode kernel mode context switching, the boundary of the security domain is the specific kernel mode interfaces of the Windows OS. Without entering into the kernel mode interfaces, any interference and tampering actions of an untrusted subject are confined within the user mode code of the process or thread running in the security context of the untrusted subject. After entering into the kernel mode interfaces, the kernel mode interfaces use the process or thread access token in the access policy enforcement decision to decide the granting of the specific access request to the untrusted subject.

### Process memory virtualization

In the case of the process memory virtualization, the security domain is the Windows OS memory manager residing in the kernel mode. As a result, without possessing additional privileges of an administrator to enter specific kernel mode interfaces successfully, an untrusted subject from the user mode is prevented from interfering and tampering the operations of the Windows OS memory manager.

### Window terminal session isolation

In the case of the window terminal session isolation, three security domains are involved as follows.

* For the session specific window manager isolation, the security domain is the Windows OS memory manager residing in the kernel mode. Without possessing additional privileges of an administrator to enter specific kernel mode interfaces successfully, an untrusted subject from the user mode is prevented from interfering and tampering the operations of the Windows OS memory manager.
* For the creation of a Windows OS process belonging to another session different from the session of the requester, the security domain is the Windows OS process manager residing in the kernel mode. Without possessing additional privileges of an administrator to enter specific kernel mode interfaces successfully, an untrusted subject from the user mode is prevented from interfering and tampering the operations of the Windows OS process manager.
* For the authorized graphical user interface level of access to a local or remote session, the security domain is the Windows OS remote window terminal service residing in a process running in the security context of the local system. Without possessing additional privileges of an administrator, the process memory virtualization prevents an untrusted subject from the user mode, representing an authenticated user, to interfere or tamper with the operations of the Windows OS remote window terminal service. In addition, the Windows OS remote window terminal service mediates the remote access to a local or remote session according to its access policy. Without being authenticated and satisfying the Windows OS remote window terminal service’s remote access policy, the network separates untrusted users from the Windows OS remote window terminal service.

### Secure desktop separation

In the case of the secure desktop separation, two security domains are involved as follows.

* For the authorized operations on the secure desktop, the security domain is the Windows OS window manager residing in the kernel mode. Without possessing additional privileges of an administrator to enter specific kernel mode interfaces successfully, an untrusted subject from the user mode is prevented from interfering and tampering the operations of the Windows OS window manager.
* For enforcing the secure desktop access policy, the security domain is the Windows OS window logon state maintaining service residing in a process running in the security context of the local system. Without possessing additional privileges of an administrator, the process memory virtualization prevents an untrusted subject from the user mode, representing an authenticated user, to interfere or tamper with the operations of the Windows OS window logon state maintaining service.

### Isolation of user-entered credential through encryption

In the case of the isolation of user-entered credential through encryption, three security domains are involved as follows.

* For encrypting the user-entered credentials, the security domain is the Windows OS window logon user interface service residing in a process running in the security context of the local system. Without possessing additional privileges of an administrator, the process memory virtualization prevents an untrusted subject from the user mode to interfere or tamper with the operations of the Windows OS window logon user interface service.
* For enforcing the security policy to gain access to the secure desktop, in which the Windows OS window logon user interface service process resides, the security domain is the Windows OS window logon state maintaining service residing in a process running in the security context of the local system. Without possessing additional privileges of an administrator, the process memory virtualization prevents an untrusted subject from the user mode to interfere or tamper with the operations of the Windows OS window logon state maintaining service.
* For the maintenance of the short-live per-boot encryption key, which is used to encrypt the user-entered credentials, the security domain is the Windows OS security driver, ksecdd.sys residing in the kernel mode. Without possessing additional privileges of an administrator to enter specific kernel mode interfaces successfully, an untrusted subject from the user mode is prevented from interfering and tampering the operations of the Windows OS security driver, ksecdd.sys.

### User interface privilege isolation within the same desktop

In the case of the user interface privilege isolation within the same desktop, the security domain is the Windows OS window manager residing in the kernel mode. Without possessing additional privileges of an administrator to enter specific kernel mode interfaces successfully, an untrusted subject from the user mode is prevented from interfering and tampering the operations of the Windows OS window manager.

### Persistent data access mediation

Recall that access to the Windows OS Active Directory, Windows OS security audit store, Windows OS WMI data store, Windows OS registry, and Windows OS NTFS volume storage locations is mediated to prevent unauthorized access.

In the case of the Windows OS registry, the security domain is the Windows OS configuration manager residing in the kernel mode. Without possessing additional privileges of an administrator to enter specific kernel mode interfaces successfully, an untrusted subject from the user mode is prevented from interfering and tampering the operations of the Windows OS configuration manager.

In the case of the Windows OS NTFS volume, the security domain is the combination of the Windows OS IO manager and the Windows OS NTFS file system (ntfs.sys), both residing in the kernel mode. Without possessing additional privileges of an administrator to enter specific kernel mode interfaces successfully, an untrusted subject from the user mode is prevented from interfering and tampering the operations of the Windows OS IO manager or the Windows OS NTFS file system.

In the case of the Windows OS Active Directory, the security domain is the LDAP server (belonging to ntdsa.dll and ntdsai.dll) residing in a process running in the security context of the local system. Without possessing additional privileges of an administrator, the process memory virtualization prevents an untrusted subject from the user mode to interfere or tamper with the operations of the LDAP server.

In the case of the Windows OS security audit store, the security domain is the RPC server listener (belonging to wevtsvc.dll) residing in a process running in the security context of the local system. Without possessing additional privileges of an administrator, the process memory virtualization prevents an untrusted subject from the user mode to interfere or tamper with the operations of the RPC server listener.

In the case of the Windows OS WMI data store, the security domain is the DCOM server listener (belonging to WMIsvc.dll) residing in a process running in the security context of the local system. Without possessing additional privileges of an administrator, the process memory virtualization prevents an untrusted subject from the user mode to interfere or tamper with the operations of the DCOM server listener.

### Encrypting file system (EFS)

In the case of the encryption file system (EFS), three security domains are involved as follows.

In the case of using a file encryption key to encrypt or decrypt file data, one of the security domains is the Windows OS NTFS file system (ntfs.sys), residing in the kernel mode. Without possessing additional privileges of an administrator to enter specific kernel mode interfaces successfully, an untrusted subject from the user mode is prevented from interfering and tampering the operations of the Windows OS NTFS file system.

In the case of using a file encryption key to encrypt or decrypt file data, the other security domain is the SMB redirector (mrxsmb20.sys), residing in the kernel mode. Without possessing additional privileges of an administrator to enter specific kernel mode interfaces successfully, an untrusted subject from the user mode is prevented from interfering and tampering the operations of the Windows OS SMB redirector.

In the case of recovering a file encryption key using a subject’s cryptographic private key, the security domain is the local EFS key management unit, residing in the Windows OS Authentication Service (lsass.exe). Without possessing additional privileges of an administrator, the process memory virtualization prevents an untrusted subject from the user mode to interfere or tamper with the operations of the Windows OS Authentication Service.

### Long lived cryptographic key isolation service

In the case of the long lived cryptographic key isolation service, the security domain is the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll). Recall that the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) is hosted in the LSASS.exe Windows OS process (i.e. the same process that also hosts the Windows OS Authentication Service). Without possessing additional privileges of an administrator, the process memory virtualization prevents an untrusted subject from the user mode to interfere or tamper with the operations of the Windows OS Authentication Service and hence the operations of the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx).

Consequently, this Commercial Grade OS Requirement Set “1.1.1.3” requirement is met.

## Addressing 1.1.1.4 “The OS shall enforce separation between the security domains of subjects”

This subsection discusses what the security domain of a subject is in a specific applicable level of protection.

### Full volume encryption

In the case of the full volume encryption, the whole Windows OS machine is deemed as a subject’s security domain. Without a valid full volume encryption PIN, the Windows OS machine is inaccessible for user logging on. Therefore, the security domain is separated from other subjects who do not possess a valid full volume encryption PIN.

### User authentication

In the case of the user authentication, defining the security domain of a subject is not necessary.

### User mode kernel mode context switching

In the case of the user mode kernel mode context switching, the security domain of a subject is the user mode virtual address space of a process associated with an access token representing the user account of the subject. Without possessing additional privileges of an administrator, the process memory virtualization prevents an untrusted subject from the user mode to interfere or tamper with the operations of the security domain of another subject.

### Process memory virtualization

In the case of the process memory virtualization, the security domain of a subject is the user mode virtual address space of a process associated with an access token representing the user account of the subject. Without possessing additional privileges of an administrator, the process memory virtualization prevents an untrusted subject from the user mode to interfere or tamper with the operations of the security domain of another subject.

### Window terminal session isolation

In the case of the window terminal session isolation, the security domain of a subject is the window terminal session where the subject belongs to. The Windows OS memory manager, the Windows OS process manager, and the Windows OS remote window terminal service enforce the window terminal session isolation security policies. Therefore, without possessing additional privileges of an administrator, an untrusted subject from the user mode is prevented from interfering or tampering with the operations of the window terminal session of another subject.

### Secure desktop separation

In the case of the secure desktop separation, defining the security domain of a subject is not necessary.

### Isolation of user-entered credential through encryption

In the case of the isolation of user-entered credential through encryption, defining the security domain of a subject is not necessary.

### User interface privilege isolation within the same desktop

In the case of the user interface privilege isolation, the integrity level of a subject defines the security domain of a subject. Subject to the user interface privilege isolation security policies enforced by the Windows OS window manager, a subject of lower integrity level is prevented from the following kinds of interfering or tampering with the security domain of a subject of higher integrity level.

* Using the handle of a window element belonging to the subject of higher integrity level for specific window manager operations.
* Sending or posting certain window messages to a window element belonging to the subject of higher integrity level, unless the window messages are explicitly exempted by the message filter associated with the subject of higher integrity level.
* Using hooks to attach to or monitor the subject of higher integrity level.
* Receiving raw keyboard or mouse data from the foreground message queue while the subject of higher integrity level is being attached to the message queue.
* Attaching a message queue of the subject of higher integrity level to a message queue of another application of a higher integrity level.
* Attaching a message queue of an application of a higher integrity level to a message queue of the subject of higher integrity level.
* Having the subject of higher integrity level to use a clip, sourced from an application of lower integrity level, from the clipboard.

Moreover, recall that the Windows OS process manager implements the following the security policy enforcement.

If the user mode code of Process A wishes to have access to a virtual memory address of Process B, then the user SID associated with the current execution thread of Process A or the user SID associated with Process A must have the necessary permissions for accessing Process B. In addition, the integrity level of Process A also must dominate the integrity level of Process B.

### Persistent data access mediation

In the case of the persistent data access mediation, defining the security domain of a subject is not necessary.

### Encrypting file system (EFS)

In the case of the encrypting file system, defining the security domain of a subject is not necessary.

### Long lived cryptographic key isolation service

In the case of the long lived cryptographic key isolation service, defining the security domain of a subject is not necessary.

Consequently, this Commercial Grade OS Requirement Set “1.1.1.4” requirement is met.

## Addressing 1.1.1.5 “The OS shall make effective use of hardware provided security features”

This subsection discusses what the hardware security feature dependency is in a specific applicable level of protection.

### Full volume encryption

As explained earlier, the full volume encryption has the following hardware security feature dependency elements.

* The [Trusted Platform Module (TPM)](http://download.microsoft.com/download/5/D/6/5D6EAF2B-7DDF-476B-93DC-7CF0072878E6/secure-start_exec.doc) unsealing command ordinal [“TPM\_ORD\_Unseal”](http://msdn.microsoft.com/en-us/library/aa376205(VS.85).aspx);
* The Memory Overwrite Request (MOR) bit available from the firmware (e.g. BIOS and EFI or PC/AT) that supports TPM.

### User authentication

The user authentication has the following optional hardware security feature dependency element.

* The smart card holding the user cryptographic private key and public key certificate for logging on through [Kerberos](http://msdn2.microsoft.com/en-us/library/aa378747(VS.85).aspx) or [RFC 2246 Transport Layer Security (TLS)](http://msdn2.microsoft.com/en-us/library/aa380123(VS.85).aspx).

### User mode kernel mode context switching

As explained earlier, the user mode kernel mode context switching has the following hardware security feature dependency elements.

* Specific instructions of the underlying hardware processor architecture to let the processor architecture reset its current privilege level;
* Limiting the use of privileged instructions to code (i.e. kernel mode code) which is executed only when the current privilege level of the processor architecture is at the highest;
* Physical memory page protection mechanisms of the processor.

### Process memory virtualization

The process memory virtualization has no hardware security feature dependency.

### Window terminal session isolation

The window terminal session isolation has no hardware security feature dependency.

### Secure desktop separation

The secure desktop separation has no hardware security feature dependency.

### Isolation of user-entered credential through encryption

The isolation of user-entered credential through encryption has no hardware security feature dependency.

### User interface privilege isolation within the same desktop

The user interface privilege isolation within the same desktop has no hardware security feature dependency.

### Persistent data access mediation

The persistent data access mediation has no hardware security feature dependency.

### Encrypting file system (EFS)

The encrypting file system has the following optional hardware security feature dependency element.

* The smart card holding the user cryptographic private key and public key certificate for recovering a file encryption key used for encrypting or decrypting an EFS-protected file.

### Long lived cryptographic key isolation service

The long lived cryptographic key isolation service has no hardware security feature dependency.

Consequently, this Commercial Grade OS Requirement Set “1.1.1.5” requirement is addressed.

We note that the “Multi-master Conflict Resolution Policy” of [Directory Replication Service (DRS)](http://msdn2.microsoft.com/en-us/library/cc203213.aspx) and the “Multi-master Conflict Resolution Policy” of [Distributed File System Replication (DFSR)](http://msdn2.microsoft.com/en-us/library/cc205266.aspx) for addressing the Commercial Grade OS Requirement Set “1.2.1.1” requirement “The OS shall ensure that security-relevant data is consistent between parts of the OS by providing a mechanism to bring inconsistent data into a consistent state in a timely manner” obviously need to depend on the real time clock provided by the underlying hardware. However, the hardware provided real time clock is not necessary deemed as a security feature of the hardware. Furthermore, we believe that this Commercial Grade OS Requirement Set “1.1.1.5” requirement is meant to seek the underlying hardware provided security features for supporting only the implementations of the separation, isolation, and least privilege security principles of the Windows OS.

Similarly, we did not mention the potential availability of the virtualization security features provided by the underlying hardware here for addressing this Commercial Grade OS Requirement Set “1.1.1.5” requirement. We did not include [Windows Server 2008 Hyper-V](http://www.microsoft.com/windowsserver2008/en/us/hyperv.aspx) as one of the Windows OS security-relevant data protection levels in the Windows OS “defense in depth” concept for addressing the Commercial Grade OS Requirement Set “1.1.1.1” requirement “The OS shall protect security-relevant data from unauthorized access”. Arguably, [Windows Server 2008 Hyper-V](http://www.microsoft.com/windowsserver2008/en/us/hyperv.aspx) desires it own treatments for addressing its corresponding unique set of security functional requirements, similar to those found in the “[U.S. Government Protection Profile for Separation Kernels in Environments Requiring High Robustness](http://www.niap-ccevs.org/cc-scheme/pp/id/pp_skpp_hr_v1.03)”.

# Meeting the “Security Architecture Separation, Isolation, and Least Privilege Management Requirements”

In the Commercial Grade OS Requirement Set, there is no individual management requirement under the heading of “Security Architecture Separation, Isolation, and Least Privilege Management Requirements”.

# Meeting the “Security Architecture Separation, Isolation, and Least Privilege Audit Requirements”

In the Commercial Grade OS Requirement Set, there is no individual audit requirement under the heading of “Security Architecture Separation, Isolation, and Least Privilege Audit Requirements”.

# Meeting the “Security Architecture Distributed Architectures Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 5 individual functional requirements under the heading of “Security Architecture Distributed Architectures Functional Requirements”. They are listed as “1.2.1.n”, where n = 1, 2, 3, 4, and 5.

## Addressing 1.2.1.1 “The OS shall ensure that security-relevant data is consistent between parts of the OS by providing a mechanism to bring inconsistent data into a consistent state in a timely manner”

When the Windows OS operates as a distributed OS over multiple computers of classes such as mobile laptops or fixed workstations and servers, the Windows OS Active Directory (AD) is necessary. The Windows OS AD stores security-relevant data of the Windows OS, and hence it is the focus on this section.

### Windows OS Active Directory (AD)

The Windows OS AD stores data objects for a Windows OS domain controller (DC) server to manage user, computer, and domain interactions, including domain-wide user logon and authentication processes, and directory searches. These data objects represent entities such as users, computers, services, and network resources. The Windows OS AD runs on multiple DC servers to provide a necessary level of scalability and availability. Therefore, objects in the Windows OS AD are distributed among all DC servers installed for a forest. A forest is a Windows OS AD concept to represent one or more domain naming contexts. Within the forest, a domain naming context provides a contiguous namespace, corresponding to a domain tree. Domain trees in the forest share a common schema, configuration, and global catalog. They exchange trust according to transitive hierarchical Kerberos trust relationships. Each tree in the forest also forms a hierarchy for the purposes of Kerberos trust.

As a result, all DC servers within the forest need to be updated so that AD object data is consistent regardless of their physical DC locations.

### Active Directory Replication

Active Directory replication is the process by which the directory data changes that originate on one DC server are automatically transferred to other DC servers that store the same data. [Directory Replication Service (DRS)](http://msdn2.microsoft.com/en-us/library/cc203213.aspx) is the mechanism that enables Active Directory replication, using RPC as its primary data transportation. DRS is implemented in dra.lib, which is statically linked to ntdsai.dll. The ntdsai.dll is loaded into ntdsa.dll. The ntdsa.dll runs in lsass.exe on a Windows OS DC server.

DRS over [SMTP](http://download.microsoft.com/download/a/e/6/ae6e4142-aa58-45c6-8dcf-a657e5900cd3/%5bMS-SRPL%5d.pdf) is also available in the Windows OS. When SMTP is used, the Ismserv.exe on the source DC uses the Collaborative Data Object (CDO) library to build an SMTP file on disk with the replication data as the attached mail message. The message file is placed in a “queue” directory. When the mail is scheduled for transfer by the mail server application, the SMTP service (Smtpsvc.dll) delivers the mail message to the destination DC over TCPIP and places the file in the “drop” directory on the destination DC. The Ismserv.exe on the destination DC then applies the updates on the destination DC.

### File Replication

While data elements of smaller sizes are stored in the Windows OS AD as AD object attribute values, data elements of larger sizes have to be stored in a local file system volume of a DC server. The System Volume (Sysvol) is a shared file directory that stores the DC server copy of public files that need to be available for authorized access. These files include logon scripts and Group Policy Object (GPO) Administrative Template .adm, .admx, and .adml files. As a result, file replication for the Sysvol directory content of all DC servers is also necessary. Both [File Replication Service (FRS)](http://msdn2.microsoft.com/en-us/library/cc205251.aspx) and [Distributed File System Replication (DFSR)](http://msdn2.microsoft.com/en-us/library/cc205266.aspx) are mechanisms that enable file replication in the Windows OS, using RPC as their data transportation. FRS supports the file replication with down level systems such as Windows 2000 or Windows Server 2003 DCs. DFSR provides a more efficient implementation and it is available on Windows Vista and Windows Server 2008. FRS is implemented in ntfrs.exe, running in the security context of local system. DFSR is implemented in dfsrS.exe, running in the security context of local system.

### Multi-master loose consistency with convergence

The replication model used in both Active Directory Replication and File Replication is called multi-master loose consistency with convergence. In this model, there are many replicas for a specific data. The replication mechanism (which is DRS, FRS, or DFSR) propagates changes made at any given replica to all other replicas. The replicas are not guaranteed to be consistent with each other at any particular time (“loose consistency”), because changes can be applied to any replica at any time (“multi-master”). If the replication mechanism is allowed to reach a steady state, in which no new updates are occurring and all previous updates have been completely replicated, all replicas are guaranteed to converge on the same set of values (“convergence”).

#### Active Directory Replication

As a LDAP server, an AD DC provides the LDAP interface with the specific LDAP commands for an authorized user to make a LDAP request to create, modify, or delete a target object, for which the AD DC is authoritative. Through the single LDAP request, multiple attributes on the object are set and modified. A LDAP request is processed as an atomic transaction. Either the entire request happens or none of it does. If the requester receives a successful response to his LDAP request, then that entire request has been committed. This is called an “originating write”. The originating write AD DC, handling the LDAP request, computes a “conflict resolution stamp” for each new or modified attribute value, and attaches this stamp to the value so when the value is replicated, the stamp is replicated too. The new stamp is unique, and in case of an update, the new stamp is greater than the stamp on the old value at that AD DC.

After a LDAP request has been committed, the originating write AD DC needs to notify its replication partner. By default, the originating write AD DC is allowed the notification interval of 15 seconds to notify its partner. However, for certain objects, the AD DC notifies its partner urgently (i.e. immediately) without pausing. For example, changes made to attributes of a user or computer account object require the urgent notification. An administrator has a couple of means to configure the notification interval. The administrator can use the [ms-DS-Replication-Notify-First-DSA-Delay](http://msdn2.microsoft.com/en-us/library/ms675144(VS.85).aspx) attribute of the Cross-Ref object. The administrator can also use the “Replicator notify pause after modify (secs)” value under the “HKLM\System\CurrentControlSet\Services\NTDS\Parameters” registry key of the originating write AD DC.

To notify a target replication partner, the originating write AD DC calls the [IDL\_DRSReplicaSync RPC](http://msdn2.microsoft.com/en-us/library/cc228241.aspx) of the target replication partner. Through the implementation of [IDL\_DRSReplicaSync RPC](http://msdn2.microsoft.com/en-us/library/cc228241.aspx), the target partner checks that the RPC caller (i.e. the originating write AD DC) has the RIGHT\_DS\_REPL\_SYNC right to notify the target partner. If the originating write AD DC has the RIGHT\_DS\_REPL\_SYNC right, then the target partner pulls the changes from the originating write AD DC by calling the originating write AD DC’s [IDL\_DRSReplicaSync RPC](http://msdn2.microsoft.com/en-us/library/cc228241.aspx) with the DRS\_INIT\_SYNC\_NOW flag.

##### Evidence indicating replication occurring

Upon calling the originating write AD DC’s [IDL\_DRSReplicaSync RPC](http://msdn2.microsoft.com/en-us/library/cc228241.aspx) with the DRS\_INIT\_SYNC\_NOW flag, the target replication partner generates the [Event ID 4932](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_REPLICA\_SOURCE\_NC\_SYNC\_BEGINS) “Synchronization of a replica of an Active Directory naming context has begun” audit record in the hard audit store to indicate the start of synchronization. This audit record includes the name of the source AD DC, the name of the destination AD DC, the naming context being replicated, and the starting update sequence number (USN).

When the synchronization completes, the target replication partner generates the [Event ID 4933](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_REPLICA\_SOURCE\_NC\_SYNC\_ENDS) “Synchronization of a replica of an Active Directory naming context has ended” audit record in the hard audit store. This audit record includes the name of the source AD DC, the name of the destination AD DC, the naming context being replicated, and the ending update sequence number (USN).

In addition, the following success or fail audit records are also generated in the hard audit store.

* [Event ID 4928](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_REPLICA\_SOURCE\_NC\_ESTABLISHED) “An Active Directory replica source naming context was established”
  + It is generated at a destination AD DC when a replica is added. This audit record includes the name of the source AD DC, the name of the destination AD DC, and the naming context of the replica being added.
* [Event ID 4929](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_REPLICA\_SOURCE\_NC\_REMOVED) “An Active Directory replica source naming context was removed”
  + It is generated at a destination AD DC when a replica is removed. This audit record includes the name of the source AD DC, the name of the destination AD DC, and the naming context of the replica being removed.
* [Event ID 4930](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_REPLICA\_SOURCE\_NC\_MODIFIED) “An Active Directory replica source naming context was modified”
  + It is generated at a destination AD DC when a replica is modified. This audit record includes the name of the source AD DC, the name of the destination AD DC, and the naming context of the replica being modified.
* [Event ID 4931](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_REPLICA\_DEST\_NC\_MODIFIED) “An Active Directory replica destination naming context was modified”
  + It is generated at a source AD DC when a replica is modified. This audit record includes the name of the source AD DC, the name of the destination AD DC, and the naming context of the replica being modified.
* [Event ID 4934](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_REPLICA\_OBJ\_ATTR\_REPLICATION) “Attributes of an Active Directory object were replicated”
  + It is generated at an AD DC when the specific attribute of a specific AD object is updated. This audit record includes the name of the AD object, the name of the attribute, and the type of change, the attribute new value, and the USN.

##### Replication Partners

As described in the above, AD replication occurs in one direction between two AD DCs at a time. The replication topology determines the replication partnerships between source and destination AD DCs.

When the Knowledge Consistency Checker (KCC) creates the replication topology, it creates connection objects on destination AD DCs that represent the inbound connections from the replication source AD DC. The KCC is implemented in ntdskcc.dll, running within lsass.exe. For each source DC that is represented by an inbound connection object, the KCC writes information to the “repsFrom” attribute of the directory partition object for each directory partition that the destination DC has in common with the source DC. This information is local to the destination DC and is not replicated. A destination DC participates in replication either by responding to notification of changes from a source DC, as described in the above, or by requesting changes to initiate replication when it starts up or in response to an administrator-configured schedule.

A source DC keeps track of its replication partners that pull changes from it and uses the information to locate partners for change notification. This information is not provided by the KCC, but rather by the source DC itself during a replication cycle. The first time a source DC receives a request for changes from a new destination DC, the source DC creates an entry for the destination DC in the “repsTo” attribute on the respective directory partition object. Whenever the source DC has changes, it calls the [IDL\_DRSReplicaSync RPC](http://msdn2.microsoft.com/en-us/library/cc228241.aspx) of all replication partners, which are identified in the “repsTo” value for the respective directory partition. Like the “repsFrom” data, this information is stored locally on the source DC and is not replicated. When updates occur, the source DC checks the “repsTo” attribute to determine the identities of its destination replication partners. The source DC notifies them one by one that changes are available.

##### Multi-master Conflict Resolution Policy

AD must ensure that all DCs agree on the value of the updated attribute after replication occurs. The general approach to conflict resolution is to order all update operations (Add, Modify, Move, and Delete) by assigning a globally unique (per-object and per-attribute) stamp to the originating update. Thus each replicated attribute value (or multi-value) is stamped during the originating update and this conflict resolution stamp is replicated with the value. The conflict resolution stamp that is applied during an originating write has the following three components.

* The version is a number that is incremented for each originating write. The version of the first originating write is 1. The version of each successive originating write is increased by 1.
* The originating time is the time of the originating write, to a one-second resolution, according to the system clock of the DC that has performed the write.
* The originating DC is the DSA GUID of the DC that has performed the originating write.

When stamps are compared, the version is the most significant, followed by the originating time and then the originating DC. If two stamps have the same version, the originating time almost always breaks the tie. In the extremely rare event that the same attribute is updated on two different DCs during the same second, the originating DC breaks the tie in an arbitrary fashion. Two different originating writes of a specific attribute of a particular object cannot assign the same stamp because each originating write advances the version at a specified originating DC. The originating time does not contribute to uniqueness. Replicated writes cannot decrease the version because values with smaller versions lose during conflict resolution.

For further information on the replication model used in Active Directory Replication, please see <http://technet2.microsoft.com/WindowsServer/en/Library/1465d773-b763-45ec-b971-c23cdc27400e1033.mspx?mfr=true>.

#### File Replication

In this section, we summarize a description of the file replication implementation of DFSR. In a pure Windows Vista and Windows Server 2008 environment, DFSR is capable to perform the replication of the Sysvol directory content among all the AD DCs. Therefore, FRS is not necessary. As result, we leave out the description of the FRS implementation in this paper.

DFSR implements a multi-master file replication system. In this system, no single computer is a master, but rather all computers in the replication system share their knowledge by exchanging version chain vectors, updates, and files. A computer may take dual roles as both a client and a server. As a client, a computer retrieves replicated metadata and replicated files from a server. Conversely, as a server, a computer serves replicated metadata and replicated files to a client.

##### Replication Partners

DFSR has two types of connections: in bound connections and outbound connections.

In an inbound connection, the DFSR service of the local server machine receives replication information from its replication partner of the connection. This replication partner is another instance of the DFSR service running on another server machine. In this case, the replication partner is called an upstream server, implementing an “Upstream Transport” object. To receive replication information from the upstream server, the local DFSR service calls the remote RPC interfaces listened by the upstream server.

In an outbound connection, the DFSR service of the local server machine provides replication information to its replication partner of the connection. In this case, the replication partner is called a downstream server, implementing a “Downstream Transport” object. To provide replication information to the downstream server, the local DFSR service listens on a set of remote RPC interfaces.

Connections are defined by the [NTDS-Connection](http://msdn.microsoft.com/en-us/library/ms683854(VS.85).aspx) objects and [ms-DFSR-Connection](http://msdn.microsoft.com/en-us/library/ms682382(VS.85).aspx) objects residing in the AD. Each connection has a “[fromServer](http://msdn.microsoft.com/en-us/library/ms675683(VS.85).aspx)” attribute where the attribute value is the distinguished name (DN) of the replication upstream server for the connection. If the “[fromServer](http://msdn.microsoft.com/en-us/library/ms675683(VS.85).aspx)” attribute value of a connection matches the DN of a server machine, then the connection is an outbound connection maintained by the local DFSR service running on the server machine.

Because the DFSR service supports [NTDS-Connection](http://msdn.microsoft.com/en-us/library/ms683854(VS.85).aspx) objects, it can provide the replication of the Sysvol (system volume) directory among AD DCs to support a distributive service such as Active Directory. As a result, when the DFSR service is running on all DCs within the deployed Windows OS environment, FRS is not necessary.

Replica set is also known as replication group. A server machine within the deployed Windows OS environment can be a member of multiple replica sets. The server machine is represented by a [computer](http://msdn.microsoft.com/en-us/library/ms680987(VS.85).aspx) object in the AD. An [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object is a member of a replica set in the Active Directory. A replica set is represented by an [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object in the AD so that the [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object has one or more [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) objects as its descendents. More than one [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) objects are possible in the AD. Each [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object has the [ms-DFSR-ComputerReference](http://msdn.microsoft.com/en-us/library/ms677139(VS.85).aspx) attribute where the value is the forward link back to a computer object. If the computer object represents a server machine, then the [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object also represents the server machine as a replica set member. Consequently, the local DFSR service on a server machine has

* specific inbound connections for certain replica sets, where it is a member;
* specific outbound connections for other replica sets, where it is a member.

##### Active Directory Polling

To bootstrap the DFSR service locally, the DFSR service reads all the available machine configuration XML file (which always exists), volume configuration XML files (if they exist locally) and replica set configuration XML files (if they exist locally). Subsequently, the DFSR service polls the AD for new configuration updates at regular intervals. The default interval value is 5 minutes.

In AD, the local machine is represented by a [computer](http://msdn.microsoft.com/en-us/library/ms680987(VS.85).aspx) object having a distinguished name. An [ms-DFSR-LocalSettings](http://msdn.microsoft.com/en-us/library/ms682399(VS.85).aspx) object can be a child of the [computer](http://msdn.microsoft.com/en-us/library/ms680987(VS.85).aspx) object. There can be more than one [ms-DFSR-LocalSettings](http://msdn.microsoft.com/en-us/library/ms682399(VS.85).aspx) child object under the [computer](http://msdn.microsoft.com/en-us/library/ms680987(VS.85).aspx) object. An [ms-DFSR-Subscriber](http://msdn.microsoft.com/en-us/library/ms682413(VS.85).aspx) object can be a child of an [ms-DFSR-LocalSettings](http://msdn.microsoft.com/en-us/library/ms682399(VS.85).aspx) object. There can be more than one [ms-DFSR-Subscriber](http://msdn.microsoft.com/en-us/library/ms682413(VS.85).aspx) child object under the [ms-DFSR-LocalSettings](http://msdn.microsoft.com/en-us/library/ms682399(VS.85).aspx) object. An ms-DFSR-Subscription object can be a child of an [ms-DFSR-Subscriber](http://msdn.microsoft.com/en-us/library/ms682413(VS.85).aspx) object. In an [ms-DFSR-Subscriber](http://msdn.microsoft.com/en-us/library/ms682413(VS.85).aspx) object, there is an “[msDFSR-MemberReference](http://msdn.microsoft.com/en-us/library/ms677157(VS.85).aspx)” attribute where the value contains the forward link to an [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object. This [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object has an “[ms-DFSR-ComputerReference](http://msdn.microsoft.com/en-us/library/ms677139(VS.85).aspx)” attribute where the value is the forward link back to the [computer](http://msdn.microsoft.com/en-us/library/ms680987(VS.85).aspx) object. Because the [computer](http://msdn.microsoft.com/en-us/library/ms680987(VS.85).aspx) object can have multiple [ms-DFSR-Subscriber](http://msdn.microsoft.com/en-us/library/ms682413(VS.85).aspx) objects, it implies that there are multiple [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) objects representing the local machine.

For each [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object representing the local machine, the following procedure is followed in order to determine the inbound and outbound connections for the local machine. The list of inbound connections for the local machine is the union of the lists of the inbound connections for the [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) objects representing the local machine. The list of outbound connections for the local machine is the union of the lists of the outbound connections for the [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) objects representing the local machine.

In AD, an [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object can be a child of an [ms-DFSR-Topology](http://msdn.microsoft.com/en-us/library/ms682422(VS.85).aspx) object, and the [ms-DFSR-Topology](http://msdn.microsoft.com/en-us/library/ms682422(VS.85).aspx) object can be a child of an [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object, and the [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object can be a child of an [ms-DFSR-GlobalSettings](http://msdn.microsoft.com/en-us/library/ms682395(VS.85).aspx) object. The [ms-DFSR-GlobalSettings](http://msdn.microsoft.com/en-us/library/ms682395(VS.85).aspx) object can be found in the AD [container](http://msdn.microsoft.com/en-us/library/ms680997(VS.85).aspx) object. In the [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object, there may be one or more [ms-DFSR-Content](http://msdn.microsoft.com/en-us/library/ms682385(VS.85).aspx) objects. Each of these [ms-DFSR-Content](http://msdn.microsoft.com/en-us/library/ms682385(VS.85).aspx) objects contains one or more [ms-DFSR-ContentSet](http://msdn.microsoft.com/en-us/library/ms682389(VS.85).aspx) objects. Every [ms-DFSR-Subscription](http://msdn.microsoft.com/en-us/library/ms682417(VS.85).aspx) object belonging to a [computer](http://msdn.microsoft.com/en-us/library/ms680987(VS.85).aspx) object has an “[msDFSR-ContentSetGuid](http://msdn.microsoft.com/en-us/library/ms677143(VS.85).aspx)” attribute where the value is a content set object GUID for identifying the applicable [ms-DFSR-Content](http://msdn.microsoft.com/en-us/library/ms682385(VS.85).aspx) objects residing under the [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object.

Every [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object contains an “[msDFSR-ReplicationGroupType](http://msdn.microsoft.com/en-us/library/ms677169(VS.85).aspx)” attribute. If the attribute value is REPLICA\_TYPE\_SYSVOL (1), then this replication group supports AD DC Sysvol directory replication. Otherwise, it supports other replication applications.

If the [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object has the REPLICA\_TYPE\_SYSVOL (1) value in its “[msDFSR-ReplicationGroupType](http://msdn.microsoft.com/en-us/library/ms677169(VS.85).aspx)” attribute, then each of its [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) objects contains the “[Server-Reference](http://msdn.microsoft.com/en-us/library/ms679773(VS.85).aspx)” attribute where the value contains the distinguished name of the DC in the domain naming context.

The list of inbound connections for this [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object is the list of [NTDS-Connection](http://msdn.microsoft.com/en-us/library/ms683854(VS.85).aspx) objects that

* reside under the DC;
* have a “[fromServer](http://msdn.microsoft.com/en-us/library/ms675683(VS.85).aspx)” attribute where the value is the distinguished name of a replication source server matching an [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object belonging to an [ms-DFSR-Topology](http://msdn.microsoft.com/en-us/library/ms682422(VS.85).aspx) object of the [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object.

The list of outbound connections for this [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object is the list of [NTDS-Connection](http://msdn.microsoft.com/en-us/library/ms683854(VS.85).aspx) objects that

* reside under the DC;
* have a parent [NTFRS-Member](http://msdn.microsoft.com/en-us/library/ms683868(VS.85).aspx) object whose distinguished name matches an [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object belonging to an [ms-DFSR-Topology](http://msdn.microsoft.com/en-us/library/ms682422(VS.85).aspx) object of the [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object.

If the [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object has the REPLICA\_TYPE\_OTHER (0), or REPLICA\_TYPE\_PROTECTION (2), or REPLICA\_TYPE\_DISTRIBUTION (3) value in its “[msDFSR-ReplicationGroupType](http://msdn.microsoft.com/en-us/library/ms677169(VS.85).aspx)” attribute, then each of its [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) objects contains a list of [ms-DFSR-Connection](http://msdn.microsoft.com/en-us/library/ms682382(VS.85).aspx) objects.

The list of inbound connections for the [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object is the list of [ms-DFSR-Connection](http://msdn.microsoft.com/en-us/library/ms682382(VS.85).aspx) objects belonging to the [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object that have a “[fromServer](http://msdn.microsoft.com/en-us/library/ms675683(VS.85).aspx)” attribute where the value is the distinguished name of a replication source server matching an [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object belonging to an [ms-DFSR-Topology](http://msdn.microsoft.com/en-us/library/ms682422(VS.85).aspx) object of the [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object.

The list of outbound connections for the [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object (MO) is the list of [ms-DFSR-Connection](http://msdn.microsoft.com/en-us/library/ms682382(VS.85).aspx) objects belonging to a [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object belonging to an [ms-DFSR-Topology](http://msdn.microsoft.com/en-us/library/ms682422(VS.85).aspx) object belonging to the [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object that have a “[fromServer](http://msdn.microsoft.com/en-us/library/ms675683(VS.85).aspx)” attribute where the value is the distinguished name of a replication source server matching the former [ms-DFSR-Member](http://msdn.microsoft.com/en-us/library/ms682403(VS.85).aspx) object (MO).

After each full Active Directory polling, the configuration updates are saved in the corresponding machine configuration XML file, volume configuration XML files and replica set configuration XML files, locally at their respective locations referenced by the corresponding registry key values. In addition, the in-memory objects: volume objects, content set Objects, replica set objects, replica member objects, and connection objects maintained by the DFSR service are also updated.

While a replica set configuration XML file is updated locally, its corresponding registry key value

* “HKLM\System\CurrentControlSet\Services\DFSR\Parameters\Replication Groups\<GUID of a Replication Group>\Replica Set Configuration File”

is updated. As this registry key value is updated, the DACL[[2]](#footnote-3) of the corresponding registry key

* “HKLM\System\CurrentControlSet\Services\Dfsr\Access Checks\Replication Groups\<GUID of a Replication Group>”

is also updated. To update this registry key DACL, the existing DACL value is read. All the non-inherited ACEs are removed from the existing value. The new DACL is appended to the remaining to form the new value of the registry key DACL. This new DACL is constructed as follows.

1. Read the security descriptor value of the “[NT-Security-Descriptor](http://msdn.microsoft.com/en-us/library/ms679006(VS.85).aspx)” attribute of the [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object representing the replica set in the AD;
2. Every DACL ACE from this security descriptor that applies to a local SID is removed;
3. Every remaining DACL ACE having ADS\_RIGHT\_DS\_READ\_PROP, ADS\_RIGHT\_DS\_WRITE\_PROP, or ADS\_RIGHT\_DS\_CONTROL\_ACCESS access is mapped into a new DACL ACE having KEY\_QUERY\_VALUE, KEY\_SET\_VALUE, or KEY\_NOTIFY respectively for the same applicable SID;
4. The mapped DACL ACEs are chained together to form the new DACL.

The registry key DACL later is used for providing access control to callers of the RPC interfaces and WMI interfaces of the local DFSR service.

##### Local data files

The name of a machine configuration XML file is stored as the data of the following registry key value.

* “HKLM\System\CurrentControlSet\Services\DFSR\Parameters\Machine Configuration File”.

The default name of the file is

* “%SystemDrive%\SYSTEM VOLUME INFORMATION\DFSR\Config\DfsrMachineConfig.XML”.

This XML file contains the following configuration attributes for the local server machine.

* CommunicationTimeoutInMs
  + Its value specifies the length of time (in milliseconds) to wait for a connection partner to respond before un-joining the connection. The default is 30\*60\*1000.
* DsPollingIntervalInMin
  + Its value specifies the time in minutes between the end of an AD polling cycle and the starting of the next. The default is 60 for a non-AD DC server.
* EnableLightDsPolling
  + Its value indicates the policy to do a lightweight AD polling with a single search to determine if a full AD polling is necessary. The default is enable.
* ShutdownTimeoutInMs
  + Its value specifies the maximum time (in milliseconds) that the DFSR service main thread will wait for all threads to exit during shutdown. The default is 90\*1000.
* RpcPortAssignment
  + Its value specifies a specific RPC port assignment. The default is to use any available port.
* LdapBindTimeoutInMs
  + Its value specifies the timeout value (in milliseconds) to wait for a LDAP bind to the Active Directory. The default is 30\*1000.
* LdapSearchTimeoutInMs
  + Its value specifies the timeout value (in milliseconds) to wait for a LDAP search on the Active Directory. The default is 10\*60\*1000.
* StagingHighWatermarkPercent
  + Its value specifies the percentage of the used staging directory size, based on the quota of a content set, before the staging directory space cleanup process is kicked off. The default is 90.
* ConflictHighWatermarkPercent
  + Its value specifies the percentage of the used conflict directory size, based on the quota of a content set, before the conflict directory space cleanup process is kicked off. The default is 90.
* ConflictLowWatermarkPercent
  + Its value specifies the percentage of the used conflict directory size needs to reach before the conflict directory space cleanup process stops. The default is 60.
* MaxOfflineTimeInDays
  + Its value specifies the number of days that a replicated folder is allowed to become offline before it is put into an error state. The default is 0.

The name of a Volume configuration XML file is stored as the data of the following registry key value.

* “HKLM\System\CurrentControlSet\Services\DFSR\Parameters\Volumes\<GUID of a Volume>\Volume Configuration File”.

The default name of the file is

* “\\?\Volume{<GUID>}\System Volume Information\DFSR\Config\Volume\_<GUID of a Volume>.XML”,

where “\\?\Volume{<GUID>}\” is mapped to a corresponding drive such as “c:\” on the local machine. This XML file contains the following configuration attributes for specific volume on the local server machine.

* VolumeGuid
  + Its value specifies the GUID of a volume on the local machine.
* SerialNumber/VolumeSerialNumber
  + Its value identifies the serial number of the (in memory) volume object. This is a unique number assigned to the in-memory volume object to represent the volume media by the local DFSR service.
* LastChangeNumber
  + Its value indicates changes in configuration between the version maintained by the DFSR WMI provider and the in memory version and\or database version of the volume configuration. The latest version has a higher LastChangeNumber value than the older version.
* LastChangeTime
  + Its value indicates the last time the volume configuration has changed. The format of the value is: MM/DD/YYYY HH:MM:SS.
* LastChangeSource
  + Its value indicates the source of the last configuration change.
* DatabasePath
  + Its value specifies the full path to the database directory. The local FRSR service creates and stores the database for the volume and the JET database logs under this directory.
* MinNtfsJournalSizeInMb
  + Its value specifies the minimum size of NTFS journal on the volume. The default is 512.
* UsnCheckPoint
  + Its value specifies the number of volume change updates to accumulate before committing a transaction to the associated database. The default is 10.
* MaxJetSessions
  + Its value specifies the maximum number of concurrent JET database sessions being opened for recording changes occurring in the volume. The default is 64.

The name of a replica set configuration XML file is stored as the data of the following registry key value.

* “HKLM\System\CurrentControlSet\Services\DFSR\Parameters\Replication Groups\<GUID of a Replication Group>\Replica Set Configuration File”.

The default name of the file is

* “%SystemDrive%\SYSTEM VOLUME INFORMATION\DFSR\Config\Replica\_<GUID of a replica set>.XML”.

This XML file contains the following configuration attributes for specific replica set on the local server machine.

* LastChangeNumber
  + Its value indicates changes in configuration between the version maintained by the DFSR WMI provider and the in memory version and\or database version of the replica set configuration. The latest version has a higher LastChangeNumber value than the older version.
* LastChangeTime
  + Its value indicates the last time the configuration has changed. The format of the value is: MM/DD/YYYY HH:MM:SS.
* LastChangeSource
  + Its value indicates the source of the last configuration change.
* Description
  + Its value specifies a description text string for describing the purpose of the replica set.
* ReplicationGroupGuid
  + Its value indicates the GUID identifier of this replica set.
* ReplicationGroupName
  + Its value indicates a user friendly name of the replica set.
* ReplicationGroupDn
  + Its value indicates the DN of the [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object representing the replica set in AD. An [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object can be a child of an [ms-DFSR-GlobalSettings](http://msdn.microsoft.com/en-us/library/ms682395(VS.85).aspx).
* ReplicationGroupType
  + Its value indicates the type of the replica set. It may be REPLICA\_TYPE\_OTHER (0), REPLICA\_TYPE\_SYSVOL (1), REPLICA\_TYPE\_PROTECTION (2), or REPLICA\_TYPE\_DISTRIBUTION (3). The default is REPLICA\_TYPE\_OTHER (0).
* TombstoneExpiryInMin
  + Its value specifies how long this replica set is kept in the database if it is deleted through replication. The default is 60\*60\*24 (i.e. 60 days).
* DefaultScheduleInUtc
  + Its value specifies whether the replication schedule value is translated in the UTC time or local time. The default is UTC time.
* DefaultSchedule
  + Its value specifies the replication schedule value.
* ReplicatedFolderGuid
  + Its value specifies the list of object GUIDs of [ms-DFSR-ContentSet](http://msdn.microsoft.com/en-us/library/ms682389(VS.85).aspx) objects which are a descendent object of an [ms-DFSR-ReplicationGroup](http://msdn.microsoft.com/en-us/library/ms682408(VS.85).aspx) object representing the replica set.

Finally, for each local volume, the volume change information such as change information of its content sets is saved to a local JET database which is stored locally on the default location of

* “\?\Volume{<GUID>}\System Volume Information\DFSR\database\_<volumeSerialNumber>\dfsr.db”.

##### Pulling Data from an upstream server

DFSR takes a three-tiered approach when pulling data from an upstream server.

* Version chain vectors are retrieved from an upstream server to determine which file versions are known to the server, but not to the downstream server. DFSR ensures that the global version sequence numbers (GVSNs) of all replicated files and file metadata that it maintains in persistent storage (that is, saved to disk) are eventually included in its version chain vector, such that the state of a server's knowledge can be determined by examining the version chain vectors alone.
* Updates, which summarize file metadata, are retrieved from an upstream server. The downstream server uses the version chain vector received from the upstream server to limit the set of updates that are retrieved from the upstream server. To retrieve all updates known to the upstream server, but not the downstream server, it is sufficient to request updates with a GVSN range over the version chain vector received from the upstream server less the version chain vector maintained by the downstream server. The updates contain file system information about the replicated files, but not the actual file data. The information includes the coordinates of the file in terms of a unique identifier (UID) identifying the file across different versions of the file, the GVSN (identifying a particular version of the file on a particular machine), a reference to the file's parent directory in terms of a UID for the parent resource (where directories are treated as files), and a file name.
* Actual file data is retrieved if a downstream server determines that the file data corresponding to a received update must be downloaded in order for the downstream server to synchronize with the upstream server.

The process of retrieving updates alternates with retrieving version chain vectors. When receiving the upstream server's version chain vector, the downstream server retrieves all updates pertaining to it, using successive calls to the upstream server.

File data can be downloaded at the same time the downstream server retrieves version chain vectors and updates. File downloads thus proceed as an independent process of synchronizing version chain vectors and updates. The downstream server specifies which file data to download based on the UID in the file metadata.

Downstream servers can update their previously saved version chain vector based on the upstream server's version chain vector after a completed synchronization; that is, when all updates pertaining to a version chain vector have been processed and all file data that a downstream server must have in order to synchronize with an upstream server has been downloaded. A downstream server's version chain vector is updated by taking the union of its version chain vector and the upstream server's version chain vector.

The version chain vectors themselves are an abstract measure of the knowledge of a member. They record the versions of files a replication partner has received, processed, and either discarded or stored in persistent storage. A replication partner can combine its version chain vector with that of an upstream partner by taking the union of the two vectors. The resulting version chain vector will also include the versions of files that the upstream partner, and by transitivity, all its partners, have processed. The difference between the version chain vectors from two replication members determines a superset of the set of updates required to synchronize one member with the contents from the other member.

##### Multi-master Conflict Resolution Policy

DFSR ensures convergence by imposing a total ordering on updates. A total ordering is obtained from the fields (consisting of: [fence, attributes, createTime, clock, uidDbGuid, uidVersion, gvsnDbGuid, and gvsnVersion](http://msdn.microsoft.com/en-us/library/cc205407.aspx)).

1. An update with a higher value of fence supersedes updates with lower fence values.
2. Otherwise, if the fence values are equal, an update with the directory attribute set in the attributes field supersedes updates that do not have the directory attribute set.
3. Otherwise, if these attributes coincide, an update with a lower value of the createTime supersedes updates with higher values.
4. Otherwise, if the create time values are equal, an update with a higher value of the clock field supersedes updates with a lower value.
5. Otherwise, if the clock fields are the same, an update with the lexicographically highest uidDbGuid supersedes one with a lower value.

* GUIDs are compared using a lexicographic left to right comparison of each byte. Where each byte is treated as an unsigned 8-bit number.

1. Otherwise, if the uidDbGuid coincide, comparison proceeds to version numbers. An update with the largest value of uidVersion supersedes an update with a lower value of a uidVersion.
2. Otherwise, if the versions are the same, an update with the lexicographically largest gvsnDbGuid supersedes one with a lower value.
3. Otherwise, if gvsnDbGuid coincide, an update with the largest gvsnVersion supersedes an update with a lower gvsnVersion.
4. Otherwise, the two updates have the same GVSN, which DFSR would allow only if the updates are in fact identical. That is, DFSR treats the fields, except for the file hashes, of an update as immutable after it is created. Furthermore, at most one machine should create an update with a given GVSN.

##### Evidence indicating replication occurring

DFSR generates the following success or fail audits in the hard audit store.

* Event ID 7002 (MSG\_EVENT\_DFSR\_AUDIT\_FILE\_SERVED) “The DFS Replication service served the following file”
  + It is generated at an upstream server when the upstream server provides a requesting downstream server over RPC a specific replica file. This audit record includes the Replicated Folder Root, the name of the Replicated Folder, the Replicated Folder ID, the File name, the File ID, the name of the downstream server.
* Event ID 7006 (MSG\_EVENT\_DFSR\_AUDIT\_UPDATE\_SENT) “The DFS Replication service sent an update for the following file”
  + It is generated at an upstream server when the upstream server provides a requesting downstream server over RPC the file data update of a specific replica file. This audit record includes the Replicated Folder Root, the name of the Replicated Folder, the Replicated Folder ID, the File name, the File ID, the name of the downstream server.
* Event ID 7004 (MSG\_EVENT\_DFSR\_AUDIT\_FILE\_DOWNLOAD) “The DFS Replication service received the following file”
  + It is generated at a downstream server when the downstream server receives a specific replica file from an upstream server over RPC. This audit record includes the Replicated Folder Root, the name of the Replicated Folder, the Replicated Folder ID, the File name, the File ID, the name of the upstream server.

Consequently, this Commercial Grade OS Requirement Set “1.2.1.1” requirement is met.

## Addressing 1.2.1.2 “The OS shall protect security-relevant data from disclosure while being transmitted to a remote part of the OS through the use of OS provided cryptographic services”

The focus on this section is the underlying transport services of DRS and DFSR. As mentioned earlier, RPC is the primary transport and SMTP is the secondary transport for DRS. For DFSR, RPC is the transport.

### RPC interface for DRS

When the local DRS server on an AD DC calls a replication partner’s [“drsuapi” RPC interface](http://msdn2.microsoft.com/en-us/library/cc228532.aspx), it needs to perform a RPC binding to the replication partner AD DC in its internal “Get Rpc Binding” function. When creating the binding with [RpcBindingSetAuthInfoExW()](http://msdn.microsoft.com/en-us/library/aa375608(VS.85).aspx),

* DRS\_RPC\_MUTUAL\_AUTH (i.e. [RPC\_C\_QOS\_CAPABILITIES\_MUTUAL\_AUTH](http://msdn.microsoft.com/en-us/library/aa378648.aspx)) is specified;
* the server principal name (SPN) of the target replication partner AD DC is constructed;
* [RPC\_C\_AUTHN\_LEVEL\_PKT\_PRIVACY](http://msdn.microsoft.com/en-us/library/aa373553(VS.85).aspx) is specified;
* [RPC\_C\_AUTHN\_GSS\_KERBEROS](http://msdn.microsoft.com/en-us/library/aa373556(VS.85).aspx) is specified.

When the local DRS server (on the target replication partner AD DC) handles an incoming request of its [“drsuapi” RPC interface](http://msdn2.microsoft.com/en-us/library/cc228532.aspx), it ensures that the caller specifies at least [RPC\_C\_AUTHN\_LEVEL\_PKT\_PRIVACY](http://msdn.microsoft.com/en-us/library/aa373553(VS.85).aspx) and that the caller is an authenticated user.

In summary, the [“drsuapi” RPC interface](http://msdn2.microsoft.com/en-us/library/cc228532.aspx) between a pair of AD replication partners is conducted over the RPC with Kerberos as the underlying security protocol, selecting the transport level encryption option for confidentiality via [RPC\_C\_AUTHN\_LEVEL\_PKT\_PRIVACY](http://msdn.microsoft.com/en-us/library/aa373553(VS.85).aspx) and the originating and target server mutual authentication option via [RPC\_C\_QOS\_CAPABILITIES\_MUTUAL\_AUTH](http://msdn.microsoft.com/en-us/library/aa378648.aspx).

### SMTP mail interface for DRS

When the local DRS server (on an originating AD DC) sends an “UPDATE REPLICA” mail to a target replication partner over SMTP in its internal “Send Upd Replica Msg” function, the mail is cryptographically signed and encrypted using [CryptSignAndEncryptMessage()](http://msdn2.microsoft.com/en-us/library/aa380278(VS.85).aspx).

During signing, the originating AD DC’s Microsoft directory service ([szOID\_NTDS\_REPLICATION “1.3.6.1.4.1.311.25.1”](http://support.microsoft.com/kb/287547)) public key certificate is used. The certificate contains a valid value of the “[objectGUID](http://msdn.microsoft.com/en-us/library/ms679021(VS.85).aspx)” attribute of an [nTDSDSA](http://msdn.microsoft.com/en-us/library/ms683855(VS.85).aspx) AD object in the local database of the originating AD DC. The originating AD DC has a valid cryptographic private key corresponding to the public key certificate for signing. By default, the hashing algorithm used is SHA256 (szOID\_RSA\_SHA256RSA “1.2.840.113549.1.1.11”). Otherwise, it uses what is specified in the “Replicator hashing algorithm (SMTP)” value under the “HKLM\System\CurrentControlSet\Services\NTDS\Parameters” registry key of the originating AD DC.

During encrypting, the target replication partner AD DC’s Microsoft directory service ([szOID\_NTDS\_REPLICATION “1.3.6.1.4.1.311.25.1”](http://support.microsoft.com/kb/287547)) public key certificate is used. The certificate contains a valid value of the “[objectGUID](http://msdn.microsoft.com/en-us/library/ms679021(VS.85).aspx)” attribute of an [nTDSDSA](http://msdn.microsoft.com/en-us/library/ms683855(VS.85).aspx) AD object in the local database of the originating AD DC. Therefore, the public key certificate is deemed as trusted, assuming correct installation of the AD. As a result, the target replication partner AD DC is authenticated. By default, the encryption algorithm used is AES128 CBC (szOID\_NIST\_AES128\_CBC “2.16.840.1.101.3.4.1.2”). Otherwise, it uses what is specified in the “Replicator encryption algorithm (SMTP)” value under the “HKLM\System\CurrentControlSet\Services\NTDS\Parameters” registry key of the originating AD DC.

When the local DRS server (on the target replication partner AD DC) handles an “UPDATE REPLICA” mail, it rejects the mail if it is not encrypted. In the local DRS server’s internal “Process Mail Msg” function, the mail is cryptographically decrypted and its message signature is cryptographically verified using [CryptDecryptAndVerifyMessageSignature()](http://msdn2.microsoft.com/en-us/library/aa379914(VS.85).aspx). The valid public key certificate used to verify the message signature must be the originate AD DC’s Microsoft directory service ([szOID\_NTDS\_REPLICATION “1.3.6.1.4.1.311.25.1”](http://support.microsoft.com/kb/287547)) public key certificate. The certificate contains a valid value of the “[objectGUID](http://msdn.microsoft.com/en-us/library/ms679021(VS.85).aspx)” attribute of an [nTDSDSA](http://msdn.microsoft.com/en-us/library/ms683855(VS.85).aspx) AD object in the local database of the target replication partner AD DC. Therefore, the public key certificate is deemed as trusted, assuming correct installation of the AD. As a result, the originate AD DC is authenticated.

In summary, the “UPDATE REPLICA” mails between a pair of AD replication partners is cryptographically signed and encrypted using trusted AD DC’s Microsoft directory service ([szOID\_NTDS\_REPLICATION “1.3.6.1.4.1.311.25.1”](http://support.microsoft.com/kb/287547)) public key certificates.

### RPC interface for DFSR

When the local DFSR server on a downstream AD DC calls an upstream AD DC’s [“FrsTransport” RPC interface](http://msdn2.microsoft.com/en-us/library/cc205338.aspx), it needs to perform a RPC binding to the upstream AD DC in its internal “Downstream Transport Setup Binding” function. When creating the binding with [RpcBindingSetAuthInfoExW()](http://msdn.microsoft.com/en-us/library/aa375608(VS.85).aspx),

* [RPC\_C\_QOS\_CAPABILITIES\_MUTUAL\_AUTH](http://msdn.microsoft.com/en-us/library/aa378648.aspx) is specified;
* the server principal name (SPN) of the target upstream AD DC is constructed;
* [RPC\_C\_AUTHN\_LEVEL\_PKT\_PRIVACY](http://msdn.microsoft.com/en-us/library/aa373553(VS.85).aspx) is specified;
* [RPC\_C\_AUTHN\_GSS\_NEGOTIATE](http://msdn.microsoft.com/en-us/library/aa373556(VS.85).aspx) is specified.

When the local DFSR server on the upstream AD DC handles an incoming request of its [“FrsTransport” RPC interface](http://msdn2.microsoft.com/en-us/library/cc205338.aspx), it ensures that the caller specifies at least [RPC\_C\_AUTHN\_LEVEL\_PKT\_PRIVACY](http://msdn.microsoft.com/en-us/library/aa373553(VS.85).aspx) and that the caller is an authenticated user.

In a pure Windows Vista and Windows Server 2008 environment, the Negotiate security provider (due to [RPC\_C\_AUTHN\_GSS\_NEGOTIATE](http://msdn.microsoft.com/en-us/library/aa373556(VS.85).aspx)) turns to the Kerberos security provider due to equal mechanism capability, as the Kerberos security provider is the primary security provider on both Windows Vista and Windows Server 2008.

In summary, the [“FrsTransport” RPC interface](http://msdn2.microsoft.com/en-us/library/cc205338.aspx) between a pair of DFSR replication partners is conducted over the RPC with Kerberos as the underlying security protocol, selecting the transport level encryption option for confidentiality via [RPC\_C\_AUTHN\_LEVEL\_PKT\_PRIVACY](http://msdn.microsoft.com/en-us/library/aa373553(VS.85).aspx) and the originating and target server mutual authentication option via [RPC\_C\_QOS\_CAPABILITIES\_MUTUAL\_AUTH](http://msdn.microsoft.com/en-us/library/aa378648.aspx).

### Crypto Systems of the Kerberos security provider

The Kerberos security provider has a fixed preferred list of crypto systems, as shown in the internal “Load CSystems” function of cryptdll.dll. At the top of the list, the “csAESk256” crypto system is specified. The “csAESk256” crypto system is defined as follows. The cryptdll.dll export CDBuildIntegrityVect() function provides the preferred crypto system list to its caller (namely, the Kerberos security provider in this case) with the KERB\_ETYPE\_AES256\_CTS\_HMAC\_SHA1\_96 Etype being the most preferred encryption type (Etype). Note that the checksum algorithm is KERB\_CHECKSUM\_HMAC\_SHA1\_96\_AES256 for the KERB\_ETYPE\_AES256\_CTS\_HMAC\_SHA1\_96 Etype.



In a pure Windows Vista and Windows Server 2008 environment, it is expected that the “csAESk256” crypto system is used, as the “csAESk256 crypto” system is the preferred crypto system on both Windows Vista and Windows Server 2008.

In the case of cryptdll.dll, it is seen from its internal “Check CNG” function that cryptdll.dll loads bcrypt.dll and uses its cryptographic services. As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista bcrypt.dll has received FIPS-140-2 (Cert # 892). Windows Server 2008 bcrypt.dll has received FIPS-140-2 (Cert # 1008).

### Cryptography used in “UPDATE REPLICA” mails

In the case of [CryptSignAndEncryptMessage()](http://msdn2.microsoft.com/en-us/library/aa380278(VS.85).aspx) and [CryptDecryptAndVerifyMessageSignature()](http://msdn2.microsoft.com/en-us/library/aa379914(VS.85).aspx), they belong to [crypt32.dll](http://msdn.microsoft.com/en-us/library/aa379884(VS.85).aspx) and live in certstor.lib. It is seen from certstor.lib that bcrypt.dll is the underlying DLL for providing the cryptographic services. As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista bcrypt.dll has received FIPS-140-2 (Cert # 892). Windows Server 2008 bcrypt.dll has received FIPS-140-2 (Cert # 1008).

Obviously, bcrypt.dll is deemed as parts of the OS provided cryptographic services.

Consequently, this Commercial Grade OS Requirement Set “1.2.1.2” requirement is met.

## Addressing 1.2.1.3 “The OS shall detect modification and insertion of security-relevant received from a remote part of the OS through the use of OS provided cryptographic services”

Recall that RPC is the underlying transport in the case of [“drsuapi” RPC interface](http://msdn2.microsoft.com/en-us/library/cc228532.aspx) and [“FrsTransport” RPC interface](http://msdn2.microsoft.com/en-us/library/cc205338.aspx), and RPC uses the Kerberos security provider.

After the Kerberos session key between a pair of AD replication partners is established, the [SealMessage()](http://msdn.microsoft.com/en-us/library/aa380181(VS.85).aspx) function and [UnsealMessage()](http://msdn.microsoft.com/en-us/library/aa380184(VS.85).aspx) function exported by the Kerberos security provider are used in the internal “Sign Or Seal” function and “Verify Or Unseal” function respectively of the RPC run time (rpcrt4.dll), running inside the originating and target replication partners.

The [SealMessage()](http://msdn.microsoft.com/en-us/library/aa380181(VS.85).aspx) function generates a cryptographic checksum of the message to be transported over RPC. The [SealMessage()](http://msdn.microsoft.com/en-us/library/aa380181(VS.85).aspx) function also includes sequencing information to prevent message loss, modification or insertion, before applying the encryption. The [UnsealMessage()](http://msdn.microsoft.com/en-us/library/aa380184(VS.85).aspx) function performs the necessary decryption before using the cryptographic checksum to detect message loss, modification or insertion.

When message loss, modification or insertion is detected, the [UnsealMessage()](http://msdn.microsoft.com/en-us/library/aa380184(VS.85).aspx) function generates the [SEC\_E\_MESSAGE\_ALTERED](http://msdn.microsoft.com/en-us/library/ms819775.aspx), [SEC\_E\_OUT\_OF\_SEQUENCE](http://msdn.microsoft.com/en-us/library/ms819775.aspx), [SEC\_E\_INVALID\_TOKEN](http://msdn.microsoft.com/en-us/library/ms819775.aspx), or [SEC\_E\_INCOMPLETE\_MESSAGE](http://msdn.microsoft.com/en-us/library/ms819775.aspx) error according to the condition for the RPC run time (rpcrt4.dll) . When the internal “Verify Or Unseal” function is told about the error, it generates the [Event ID 4816](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_RPC\_INTEGRITY\_VIOLATION/SE\_AUDITID\_ETW\_RPC\_INTEGRITY\_VIOLATION) “RPC detected an integrity violation while decrypting an incoming message” audit record in the hard audit store. This audit record includes the name of the peer who is the originator of the message.

In the case of DRS over [SMTP](http://download.microsoft.com/download/a/e/6/ae6e4142-aa58-45c6-8dcf-a657e5900cd3/%5bMS-SRPL%5d.pdf), we recall that [CryptDecryptAndVerifyMessageSignature()](http://msdn2.microsoft.com/en-us/library/aa379914(VS.85).aspx) is used when processing an incoming “UPDATE REPLICA” mail. In the local DRS server’s internal “Process Mail Msg” function, if the message’s integrity cannot be verified, the mail is rejected. An Event ID 1173 (DIRLOG\_EXCEPTION) “Internal event: Active Directory Domain Services has encountered the following exception and associated parameters” soft audit event record is generated to the soft audit storage. This soft audit event record includes the error code [ERROR\_ENCRYPTION\_FAILED](http://msdn.microsoft.com/en-us/library/ms819773.aspx).

The soft audit storage is the audit storage that does not cause the local machine to prevent subsequent auditable events, except those taken by an administrator, to occur when the audit storage is full (i.e. reaching its maximum size).

Consequently, this Commercial Grade OS Requirement Set “1.2.1.3” requirement is met.

## Addressing 1.2.1.4 “Upon detection of modification and insertion of security-relevant received from a remote part of the OS, the OS shall reject the data”

When the internal “Verify Or Unseal” function of the RPC run time (rpcrt4.dll) is told the [SEC\_E\_MESSAGE\_ALTERED](http://msdn.microsoft.com/en-us/library/ms819775.aspx), [SEC\_E\_OUT\_OF\_SEQUENCE](http://msdn.microsoft.com/en-us/library/ms819775.aspx), [SEC\_E\_INVALID\_TOKEN](http://msdn.microsoft.com/en-us/library/ms819775.aspx), or [SEC\_E\_INCOMPLETE\_MESSAGE](http://msdn.microsoft.com/en-us/library/ms819775.aspx) error, the corresponding RPC message, having been verified by the [UnsealMessage()](http://msdn.microsoft.com/en-us/library/aa380184(VS.85).aspx) function, is not processed further. The error [RPC\_S\_SEC\_PKG\_ERROR](http://msdn.microsoft.com/en-us/library/aa378645(VS.85).aspx) is generated. The RPC handers, defined in the [“drsuapi” RPC interface](http://msdn2.microsoft.com/en-us/library/cc228532.aspx) and [“FrsTransport” RPC interface](http://msdn2.microsoft.com/en-us/library/cc205338.aspx) inside the DRS server and the DFSR server, are not even invoked.

In the case of DRS over [SMTP](http://download.microsoft.com/download/a/e/6/ae6e4142-aa58-45c6-8dcf-a657e5900cd3/%5bMS-SRPL%5d.pdf), we recall that [CryptDecryptAndVerifyMessageSignature()](http://msdn2.microsoft.com/en-us/library/aa379914(VS.85).aspx) is used when processing an incoming “UPDATE REPLICA” mail. In the local DRS server’s internal “Process Mail Msg” function, if the message’s integrity cannot be verified, the mail is rejected.

Consequently, this Commercial Grade OS Requirement Set “1.2.1.4” requirement is met.

## Addressing 1.2.1.5 “The OS shall provide a means for secure remote administration using OS provided cryptographic services”

In the case of DRS, its management interface is also included in the [“drsuapi” RPC interface](http://msdn2.microsoft.com/en-us/library/cc228532.aspx). As explained earlier, the RPC transport of the [“drsuapi” RPC interface](http://msdn2.microsoft.com/en-us/library/cc228532.aspx) is secure with mutual authentication, encryption, and integrity protection with the use of FIPS-140-2 validated cryptographic services. Therefore, the DRS management interface is also secure with the use of FIPS-140-2 validated cryptographic services.

In the case of DFSR, its management interface is provided through WMI as dfsrS.exe is also a WMI provider with the [“dfsrprovs.mof”](http://msdn2.microsoft.com/en-us/library/bb540028.aspx) file, using the WMI “\\.\Root\MicrosoftDfs” or “\\<TragetServerName>\Root\MicrosoftDfs” namespace. The access to the management service of the provider is provided by the WMI infrastructure. As described earlier, the WMI infrastructure is a DCOM server. A DCOM server runs over the RPC run time (rpcrt4.dll) also. Therefore the security for RPC that we discussed earlier applies to the DFSR management interface through WMI also. As a result, the DFSR management interface is also secure with the use of FIPS-140-2 validated cryptographic services.

## Addressing 1.2.1.6 “The OS shall authenticate remote parts of the OS using OS provided cryptographic services”

As explained earlier, the [“drsuapi” RPC interface](http://msdn2.microsoft.com/en-us/library/cc228532.aspx) between a pair of AD replication partners is conducted over the RPC with Kerberos as the underlying security protocol, selecting the transport level encryption option for confidentiality via [RPC\_C\_AUTHN\_LEVEL\_PKT\_PRIVACY](http://msdn.microsoft.com/en-us/library/aa373553(VS.85).aspx) and the originating and target server mutual authentication option via [RPC\_C\_QOS\_CAPABILITIES\_MUTUAL\_AUTH](http://msdn.microsoft.com/en-us/library/aa378648.aspx).

Similarly, the [“FrsTransport” RPC interface](http://msdn2.microsoft.com/en-us/library/cc205338.aspx) between a pair of DFSR replication partners is conducted over the RPC with Kerberos as the underlying security protocol, selecting the transport level encryption option for confidentiality via [RPC\_C\_AUTHN\_LEVEL\_PKT\_PRIVACY](http://msdn.microsoft.com/en-us/library/aa373553(VS.85).aspx) and the originating and target server mutual authentication option via [RPC\_C\_QOS\_CAPABILITIES\_MUTUAL\_AUTH](http://msdn.microsoft.com/en-us/library/aa378648.aspx).

Finally, the “UPDATE REPLICA” mails between a pair of AD replication partners are cryptographically signed and encrypted using trusted originate AD DCs’ Microsoft directory service ([szOID\_NTDS\_REPLICATION “1.3.6.1.4.1.311.25.1”](http://support.microsoft.com/kb/287547)) public key certificates.

Consequently, this Commercial Grade OS Requirement Set “1.2.1.6” requirement is met.

# Meeting the “Security Architecture Distributed Architectures Management Requirements”

In the Commercial Grade OS Requirement Set, there is one individual management requirement under the heading of “Distributed Architectures Management Requirements”. It is listed as “1.2.2.1”.

## Addressing 1.2.2.1 “The OS shall provide the ability for an authorized administrator to remotely manage the OS”

In a distributed environment with many managed computers (including servers, fixed workstations and mobile laptop), the ideal methodology to manage the environment, maintained by the Windows OS, is the [Windows OS group policy](http://msdn2.microsoft.com/en-us/library/aa374177.aspx).

### Windows OS group policy

The Windows OS group policy enables policy-based administration that uses the Windows OS AD. The Windows OS group policy uses directory services and security group membership to provide flexibility and support extensive configuration information. Policy settings are specified by an administrator. Due to the availability of

* the LDAP interface provided by the Windows OS AD;
* the remote file system services over SMB V2 provided by the Windows OS Server drivers and the Windows OS SMB remote file system redirector to the replicated and shared Sysvol file directory of an AD DC server,

centralized policy-based administration enables an administrator to control at least the following computer and user management settings through the Windows OS group policy infrastructure.

* [Registry-based policy settings](http://msdn.microsoft.com/en-us/library/aa371968(VS.85).aspx)
  + Specify registry-based settings using the Administrative Templates node of the Group Policy Object Editor.
* [Security settings](http://msdn.microsoft.com/en-us/library/aa371968(VS.85).aspx)
  + Define security settings for the local computer, domain, and network.
* [Software installation](http://msdn.microsoft.com/en-us/library/aa371968(VS.85).aspx)
  + Deploy applications as either assigned (i.e. the administrator mandates the installation) or published (i.e. the administrator mandates provides applications that users can choose to install). Update or remove applications.
* [Scripts](http://msdn.microsoft.com/en-us/library/aa371968(VS.85).aspx)
  + Specify scripts to run at computer startup and operating system shutdown, and when a user logs on or logs off.
* [Remote Installation Services](http://msdn.microsoft.com/en-us/library/aa371968(VS.85).aspx)
  + Control the behavior of the remote installation feature, as displayed to managed computers.
* [Internet Explorer maintenance](http://msdn.microsoft.com/en-us/library/aa371968(VS.85).aspx)
  + Manage and customize Microsoft Internet Explorer on managed computers running the Windows OS.
* [Internet Explorer 7](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Replace and expand the current settings in the Internet Explorer Maintenance extension to allow administrators the ability to read the current settings without affecting values.
* [Folder redirection](http://msdn.microsoft.com/en-us/library/aa371968(VS.85).aspx)
  + Redirect Shell special folders to specific network server locations.
* [Shell First Experience, Logon, and Privileges](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Configure the logon experience to include expanded Group Policy settings in:
    - Roaming User Profiles.
    - Redirected folders.
    - Logon dialog screens.
* [Antivirus](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Manage behavior for evaluating high-risk attachments.
* [Background Intelligent Transfer Service (BITS)](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Configure the BITS Neighbor Casting feature (new in Windows Vista and Windows Server 2008) to facilitate peer-to-peer file transfer within a domain.
* [Client Help](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Determine where users access Help systems that may include untrusted content. The administrator can direct users to Help or to local offline Help.
* [Deployed Printer Connections](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Deploy a printer connection to a computer. This is useful when the computer is shared in a locked-down environment, such as when a user roams to a different location and needs to have a printer connected automatically.
* [Device Installation](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Allow or deny a device installation, based upon the device class or ID.
* [Disk Failure Diagnostic](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Control the level of information displayed by the disk failure diagnostic.
* [DVD Video Burning](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Customize the video disc authoring experience.
* Enterprise Quality of Service (QoS)
  + Alleviate network congestion issues by enabling central management of Windows Server 2008 network traffic. Without requiring changes to applications, the administrator defines flexible policies to prioritize the Differentiated Services Code Point (DSCP) marking and throttle rate.
* [Hybrid Hard Disk](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Configure the hybrid hard disk (with non-volatile cache) properties, allowing the administrator to manage:
    - Use of non-volatile cache.
    - Startup and resume optimizations.
    - Solid state mode.
    - Power savings mode.
* [Networking: Quarantine](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Manage the following three components:
    - Health Registration Authority (HRA)
    - Internet Authentication Service (IAS)
    - Network Access Protection (NAP)
* [Networking: Wired Wireless](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Apply a generic architecture for centrally managing existing and future media types.
* [Power Management](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Configure any current power management options in the Control Panel.
* [Removable Storage](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Protect storage data by limiting the data that can be read from and written to removable storage devices. Administrators enforce restrictions on specific computers or users without relying on third party products or disabling the buses.
* [Security Protection](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Combine the management of both the Windows Firewall and IPSEC technologies to reduce the possibility of creating conflicting rules. Administrators specify which applications or ports to open and whether or not connections to those resources must be secure.
* [Shell Application Management](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Manage access to the toolbar, taskbar, Start menu, and icon displays.
* [Shell Sharing, Sync, and Roaming](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Customize:
    - Autorun for different devices and media.
    - Creation and removal of partnerships.
    - Synchronization schedule and behavior.
    - Creation and access to workspaces.
* [Shell Visuals](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Configure the desktop display to include:
    - AERO Glass display.
    - New screen saver behavior.
    - Search and views.
* [Tablet PC](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Configure Tablet PC to include:
    - Tablet Ink Watson and Personalization features.
    - Tablet PC desktop features.
    - Input Panel features.
    - Tablet PC touch input.
* [Terminal Services](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Configure the following features to enhance the security, ease-of-use, and manageability of Terminal Services remote connections.
    - Allow or prevent redirection of additional supported devices to the remote computer in a Terminal Services session.
    - Require the use of CredSSP, Transport Layer Security (TLS) 1.0 or native Remote Desktop Protocol (RDP) encryption, or negotiate a security method.
    - Require the use of a specific encryption level (FIPS Compliant, High, Client Compatible, or Low).
* [Troubleshooting and Diagnostics](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Control the diagnostic level from automatically detecting and fixing problems to indicating to the user that assisted resolution is available for:
    - Application issues.
    - Leak detection.
    - Resource allocation.
* [User Account Protection](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Configure the properties of user accounts to:
    - Determine behavior for the elevation prompt.
    - Elevate the user account during application installs.
    - Identify the least-privileged user accounts.
    - Virtualize file and registry write failures to per-user locations.
* [Windows Error Reporting](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc)
  + Disable Windows Feedback only for Windows or for all components. By default, Windows Feedback is turned on for all Windows components.

The administrator can apply these settings to groups of computers or users using the infrastructure provided by the Windows OS AD. The administrator can manage these settings from a single location through the LDAP interface of the Windows OS AD, without physically touching individual computers in the distributed environment maintained by the Windows OS.

### Group policy processing at a local computer

At a high level, there are two phases to Group Policy processing:

* Group Policy Infrastructure processing;
* Client-Side Extension (CSE) processing.

In the Group Policy Infrastructure processing phase, the Windows OS group policy client service (gpsvc.dll) residing on a managed computer queries its closest AD DC server over LDAP to determine

* what the link speed to the DC is;
* where the managed computer lives in the AD hierarchy (that is, which site, domain, and organizational unit (OU) the managed computer is a member of);
* which GPOs apply to the computer or a currently logged-on user.

The Windows OS group policy client service runs in the security context of the local system on the managed computer.

Once the list of GPOs has been built in the Group Policy Infrastructure processing phase, the Client-Side Extension (CSE) processing phase is entered as the next phase. During the CSE processing phase, each [registered CSE processes the list of GPOs that have implemented settings in its applicable area](http://msdn.microsoft.com/en-us/library/aa373489(VS.85).aspx). For example, the Registry or Administrative Template CSE runs first in all cases. It processes all GPOs that apply to the managed computer or a currently logged-on user, and that have implemented registry policy within them.

The list that follows details the steps where the Group Policy Infrastructure processing cycle goes through. These steps include network interactions between the managed computer and an AD DC server. The Windows OS group policy client service on the managed computer actually performs the processing cycle simultaneously for both the managed computer and a currently logged-on user, with each cycle running on a different thread within the Windows OS group policy client service process.

1. The local Windows OS group policy client service on the managed computer uses the local [Network Location Awareness (NLA)](http://msdn.microsoft.com/en-us/library/ms740558.aspx) service to determine/estimate the link speed of the connection to an AD DC server in its site.
2. The local Windows OS group policy client service reads CSE status information from its local registry to determine which GPOs were processed last.
3. The local Windows OS group policy client service uses LDAP to search the “[gpLink](http://msdn2.microsoft.com/en-us/library/ms675727.aspx)” attribute in the AD on each [container](http://msdn.microsoft.com/en-us/library/ms680997(VS.85).aspx) object of the managed computer’s location in the AD hierarchy, by looking first at the OU level (including all nested OUs), then at the domain, and finally at the AD site level. From the results of this search, it builds a list of GPOs that must be evaluated for processing.
4. Each GPO is then searched in AD to determine whether the managed computer or a currently logged-on user has the necessary permissions to process it. Its version number, the path to the [Group Policy Template (GPT)](http://msdn.microsoft.com/en-us/library/aa374180(VS.85).aspx) portion of the GPO in the replicated SYSVOL directory of the AD DC server, and the specific CSEs implemented in that GPO are also evaluated.
5. Using the remote file system services over SMB V2 provided by the local Windows OS SMB remote file system redirector, the local Windows OS group policy client service reads the contents of the GPT and get the GPO's version number from the “gpt.ini” file. The version numbers in the [Group Policy Container (GPC)](http://msdn.microsoft.com/en-us/library/ms682264(VS.85).aspx) and GPT are one factor that is used to determine whether a GPO has changed since the last processing cycle.
6. Each CSE runs in the order that is registered under the “HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\GPExtensions” registry key, and processes the GPOs that implement that CSE if the GPO has changed since last processing cycle. Each CSE also logs [Resultant Set of User Policy (RSOP)](http://msdn.microsoft.com/en-us/library/aa374408(VS.85).aspx) data to the local Windows Management Instrumentation (WMI) during each refresh, if available.

Foreground GPO processing occurs for the managed computer during its system restart and for its users during a user logon. Background GPO processing refreshes occur on the managed computer by default every 90 minutes plus up to 30 minutes in a randomized value. Background GPO processing refreshes occur on AD DC servers every 5 minutes by default. There are also NLA-based GPO processing refreshes, which are essentially background GPO processing refresh events that are triggered by a previous failure of Group Policy processing due to the lack of access to an AD DC server (as when the managed computer was offline during its system restart).

A GPO is composed of two pieces:

* the [GPC](http://msdn.microsoft.com/en-us/library/ms682264(VS.85).aspx) (as a AD object) residing in AD under the “CN=Policies, CN=System” container within each domain;
* the [GPT](http://msdn.microsoft.com/en-us/library/aa374180(VS.85).aspx) (as a file system directory) residing under the “Policies” directory within in the replicated SYSVOL directory.

Each piece of the GPO contains a version number. For the [GPC](http://msdn.microsoft.com/en-us/library/ms682264(VS.85).aspx), this version number is stored in the “[versionNumber](http://msdn.microsoft.com/en-us/library/ms680897(VS.85).aspx)” attribute of the [GPC](http://msdn.microsoft.com/en-us/library/ms682264(VS.85).aspx) object. For the [GPT](http://msdn.microsoft.com/en-us/library/aa374180(VS.85).aspx), its version number is stored within the “gpt.ini” file at the root of a given [GPT](http://msdn.microsoft.com/en-us/library/aa374180(VS.85).aspx). The local Windows OS group policy client service on the managed computer also keeps a record of the version numbers of the GPOs it has processed (for both per-computer and per-logged on user) within the local registry. This version information is held under the registry keys:

* “HKLM\Software\Microsoft\Windows\Currentversion\Group Policy\History” for the computer;
* “HKLM\Software\Microsoft\Windows\Currentversion\Group Policy\<SID of User>” for the user logged on to the managed computer.

When a Group Policy processing occurs, one of its tasks is to examine the version numbers of all GPOs that the managed computer or the logged-on user are subject to and compare them to any, previously processed during the last cycle, as found in the registry. If any of the version numbers of the current GPOs are different, those GPOs will be processed during the current processing cycle. If not, they are not processed unless one of the other change conditions is met. Those other change conditions are as follows:

* a change in the list of GPOs that apply to a user or computer (a GPO has been added or removed);
* a change in the security group membership of a currently logged-on user or the managed computer;
* a change in a [WMI filter linked to a GPO](http://technet.microsoft.com/en-us/library/cc787382.aspx) (a WMI filter has been added or removed).

### Transportation security of the LDAP connection to AD

As mentioned earlier, LDAP is a transport protocol, which uses one of the Windows OS supported security providers. When the Windows OS group policy client service (gpsvc.dll) uses [ldap\_bind\_s()](http://msdn2.microsoft.com/en-us/library/aa366156.aspx) to bind to the AD, it specifies the Kerberos security provider. Additionally, it uses [ldap\_set\_option()](http://msdn2.microsoft.com/en-us/library/aa366993(VS.85).aspx) to require message signing and verification by specifying [LDAP\_OPT\_SIGN](http://msdn.microsoft.com/en-us/library/aa366993(VS.85).aspx). The LDAP client library is implemented in wldap32.dll running in the same Windows OS process address space as the Windows OS group policy client service (gpsvc.dll). When the LDAP client library (wldap32.dll) handles [ldap\_bind\_s()](http://msdn2.microsoft.com/en-us/library/aa366156.aspx) using the Kerberos security provider, it also uses the [ISC\_REQ\_MUTUAL\_AUTH](http://msdn.microsoft.com/en-us/library/aa375507(VS.85).aspx) flag when calling the [InitializeSecurityContext()](http://msdn.microsoft.com/en-us/library/aa375507(VS.85).aspx) exposed by the Kerberos security provider. The [ISC\_REQ\_MUTUAL\_AUTH](http://msdn.microsoft.com/en-us/library/aa375507(VS.85).aspx) flag ensures the authentication of the AD DC as the LDAP server being bound to.

To verifying the message signature, the LDAP client library (wldap32.dll) uses the [UnsealMessage()](http://msdn.microsoft.com/en-us/library/aa380184(VS.85).aspx) function exported by the Kerberos security provider (Kerberos.dll). If the [UnsealMessage()](http://msdn.microsoft.com/en-us/library/aa380184(VS.85).aspx) function returns a failure, the message is dropped. Consequently, the Windows OS group policy client service (gpsvc.dll) does not receive any valid data to use in its requests of LDAP operations.

In summary, the LDAP connection between the Windows OS group policy client service (gpsvc.dll) (residing on a managed computer) and the Windows OS AD is secure with

* the mutual authentication;
* the message integrity.

### Transportation security of the SMB V2 connection to the replicated SysVol directory of an AD DC

As mentioned earlier, SMB V2 is a transport protocol, which uses one of the Windows OS supported security providers. For an outgoing SMB V2 connection, its service is provided by the local Windows OS SMB remote file system redirector (mrxsmb20.sys). The Windows OS group policy client service (gpsvc.dll) gets the redirector’s service transparently through the local IO manager. The Windows OS group policy client service (gpsvc.dll) only needs to call [CreateFile()](http://msdn2.microsoft.com/en-us/library/aa363858.aspx) by specifying the full path name to the actual file location on the target remote server.

The connection security behavior provided by the Windows OS SMB remote file system redirector (mrxsmb20.sys) is uniform for all its local users.

In its internal “Build Session Setup Security Blob” function, the redirector always specifies [ISC\_REQ\_MUTUAL\_AUTH](http://msdn.microsoft.com/en-us/library/aa375507(VS.85).aspx) to ensure the target server authenticity. It constructs the server principal name (SPN) of the target remote file server based on its client’s supplied information such as a target server name. It specifies the [Negotiate security provider](http://msdn.microsoft.com/en-us/library/aa373556(VS.85).aspx). In a pure Windows Vista and Windows Server 2008 environment, the Negotiate security provider turns to the Kerberos security provider due to equal mechanism capability, as the Kerberos security provider is the primary security provider on both Windows Vista and Windows Server 2008.

By default, the Windows OS SMB remote file system redirector (mrxsmb20.sys) always specifies [SMB2\_NEGOTIATE\_SIGNING\_ENABLED](http://msdn.microsoft.com/en-us/library/cc246624.aspx) in the [SMB2\_REQ\_NEGOTIATE](http://msdn.microsoft.com/en-us/library/cc212892.aspx) structure, when negotiating the SMB V2 capability with the Windows OS Server driver (srv2.sys) of the target remote file server over SMB V2. By default, the Windows OS Server driver also supports [SMB2\_NEGOTIATE\_SIGNING\_ENABLED](http://msdn.microsoft.com/en-us/library/cc246624.aspx). In addition, by default, the Windows OS Server driver (srv2.sys) always sets [SMB2\_NEGOTIATE\_SIGNING\_REQUIRED](http://msdn.microsoft.com/en-us/library/cc212693.aspx) in its SMB V2 negotiate response. Therefore, in a pure Windows Vista and Windows Server 2008 environment, the [SMB2\_NEGOTIATE\_SIGNING\_REQUIRED](http://msdn.microsoft.com/en-us/library/cc212693.aspx) is always a negotiated outcome between the Windows OS SMB remote file system redirector (mrxsmb20.sys) and the Windows OS Server driver (srv2.sys).

Due to [SMB2\_NEGOTIATE\_SIGNING\_REQUIRED](http://msdn.microsoft.com/en-us/library/cc212693.aspx) as the negotiated outcome, outgoing SMB V2 messages from the redirector are signed in its internal “Compute Outgoing Signature” function, and incoming SMB V2 messages to the redirector are verified for their signatures with its internal “Validate Signature” function. The cryptographic hashing algorithm used in the signature verification is SHA256 HMAC provided by the FIPS-140-2 validated ksecdd.sys, using the session key established through the Kerberos protocol. In the case where the message signature is not verified, the message is dropped in the redirector’s internal “Receive Indication Handler” function.

In summary, the SMB V2 connection between the Windows OS group policy client service (gpsvc.dll) (residing on a managed computer) and the replicated SysVol directory of a Windows OS AD DC is secure with

* the mutual authentication;
* the message integrity.

Consequently, this Commercial Grade OS Requirement Set “1.2.2.1” requirement and the Commercial Grade OS Requirement Set “1.2.1.5” requirement are satisfied.

# Meeting the “Security Architecture Distributed Architectures Audit Requirements”

In the Commercial Grade OS Requirement Set, there is one individual audit requirement under the heading of “Distributed Architectures Audit Requirements”. It is listed as “1.2.3.1”.

## Addressing 1.2.3.1 “The OS shall provide the ability to audit when modification or insertion of security-relevant data received from a remote part of the OS has been detected”

The below behaviors of DRS and DFSR have been described earlier.

When DRS or DFRS RPC message loss, modification or insertion is detected, the [UnsealMessage()](http://msdn.microsoft.com/en-us/library/aa380184(VS.85).aspx) function generates the [SEC\_E\_MESSAGE\_ALTERED](http://msdn.microsoft.com/en-us/library/ms819775.aspx), [SEC\_E\_OUT\_OF\_SEQUENCE](http://msdn.microsoft.com/en-us/library/ms819775.aspx), [SEC\_E\_INVALID\_TOKEN](http://msdn.microsoft.com/en-us/library/ms819775.aspx), or [SEC\_E\_INCOMPLETE\_MESSAGE](http://msdn.microsoft.com/en-us/library/ms819775.aspx) error according to the condition for the RPC run time (rpcrt4.dll) . When the internal “Verify Or Unseal” function is told about the error, it generates the [Event ID 4816](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_RPC\_INTEGRITY\_VIOLATION/SE\_AUDITID\_ETW\_RPC\_INTEGRITY\_VIOLATION) “RPC detected an integrity violation while decrypting an incoming message” audit record in the hard audit store. This audit record includes the name of the peer who is the originator of the message.

In the case of DRS over [SMTP](http://download.microsoft.com/download/a/e/6/ae6e4142-aa58-45c6-8dcf-a657e5900cd3/%5bMS-SRPL%5d.pdf), we recall that [CryptDecryptAndVerifyMessageSignature()](http://msdn2.microsoft.com/en-us/library/aa379914(VS.85).aspx) is used when processing an incoming “UPDATE REPLICA” mail. In the local DRS server’s internal “Process Mail Msg” function, if the message’s integrity cannot be verified, the mail is rejected. An Event ID 1173 (DIRLOG\_EXCEPTION) “Internal event: Active Directory Domain Services has encountered the following exception and associated parameters” soft audit event record is generated to the soft audit storage. This soft audit event record includes the error code [ERROR\_ENCRYPTION\_FAILED](http://msdn.microsoft.com/en-us/library/ms819773.aspx).

The soft audit storage is the audit storage that does not cause the local machine to prevent subsequent auditable events, except those taken by an administrator, to occur when the audit storage is full (i.e. reaching its maximum size).

Consequently, this Commercial Grade OS Requirement Set “1.2.3.1” requirement is met.

# Meeting the “Access Control Polices Discretionary Access Control Policy Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 6 individual functional requirements under the heading of “Access Control Policies Discretionary Access Control Policy Functional Requirements”. They are listed as “2.1.1.n”, where n = 1, 2, 3, 4, 5, and 6.

## Addressing 2.1.1.1 “The OS shall enforce a Discretionary Access Control policy on all subjects, all named objects and all access operations among them”

The following tables (Table 1 and Table 2) list the Windows OS named objects, to which the Windows OS Discretionary Access Control (DAC) policy is applicable. The standard access control elements applicable to every named object are:

* [DELETE](http://msdn.microsoft.com/en-us/library/aa379607.aspx);
* [READ\_CONTROL](http://msdn.microsoft.com/en-us/library/aa379607.aspx);
* [WRITE\_DAC](http://msdn.microsoft.com/en-us/library/aa379607.aspx);
* [WRITE\_OWNER](http://msdn.microsoft.com/en-us/library/aa379607.aspx);
* [SYNCHRONIZE](http://msdn.microsoft.com/en-us/library/aa379607.aspx).

In the last column of the tables, the object specific access control elements defined for each named object “class” are provided, in additional to the standard access control elements.

Table 1: Windows OS kernel mode named objects and their specific access control elements

|  |  |  |  |
| --- | --- | --- | --- |
| **Named object** | **Resource manager** | **How it is referenced** | **Supported object specific access control elements** |
| Windows OS EtwRegistration GUID objects | Windows OS Event Tracing Manager (etw.lib) | ETW channel/session GUID | WMIGUID\_QUERY WMIGUID\_SET WMIGUID\_NOTIFICATION WMIGUID\_READ\_DESCRIPTION WMIGUID\_EXECUTE TRACELOG\_CREATE\_REALTIME TRACELOG\_CREATE\_ONDISK TRACELOG\_GUID\_ENABLE TRACELOG\_ACCESS\_KERNEL\_LOGGER TRACELOG\_LOG\_EVENT TRACELOG\_CREATE\_INPROC TRACELOG\_ACCESS\_REALTIME TRACELOG\_REGISTER\_GUIDS |
| Windows OS kernel mode window management infrastructure (WMI) GUID objects | Windows OS kernel mode window management infrastructure (WMI) (wmi.lib) | WMI GUID | WMIGUID\_QUERY WMIGUID\_SET WMIGUID\_EXECUTE |
| Windows OS IO manager Device objects | Windows OS IO Manager (io.lib) | Device name | FILE\_READ\_ACCESS FILE\_WRITE\_ACCESS |
| Windows OS IO manager Driver objects | Windows OS IO Manager (io.lib) | Driver name | FILE\_READ\_DATA FILE\_WRITE\_DATA FILE\_EXECUTE |
| Windows OS IO manager IoCompletion objects | Windows OS IO Manager (io.lib) | IoCompletion name | IO\_COMPLETION\_QUERY\_STATE IO\_COMPLETION\_MODIFY\_STATE |
| Windows OS IO manager device file objects | Windows OS IO Manager (io.lib) | Device file name | FILE\_READ\_DATA FILE\_READ\_ATTRIBUTES FILE\_READ\_EA FILE\_WRITE\_DATA FILE\_WRITE\_ATTRIBUTES FILE\_WRITE\_EA FILE\_APPEND\_DATA FILE\_EXECUTE |
| File system volume objects | Windows OS IO Manager (io.lib) | Volume path name | FILE\_READ\_DATA FILE\_WRITE\_DATA FILE\_READ\_ATTRIBUTES FILE\_WRITE\_ATTRIBUTES |
| Windows OS Process objects | Windows OS Process Manager (ps.lib) | Process ID | PROCESS\_TERMINATE  PROCESS\_CREATE\_THREAD  PROCESS\_SET\_SESSIONID  PROCESS\_VM\_OPERATION PROCESS\_VM\_READ  PROCESS\_VM\_WRITE  PROCESS\_DUP\_HANDLE PROCESS\_CREATE\_PROCESS  PROCESS\_SET\_QUOTA  PROCESS\_SET\_INFORMATION  PROCESS\_QUERY\_INFORMATION  PROCESS\_SUSPEND\_RESUME  PROCESS\_QUERY\_LIMITED\_INFORMATION |
| Windows OS Thread objects | Windows OS Process Manager (ps.lib) | Thread ID | THREAD\_TERMINATE THREAD\_SUSPEND\_RESUME  THREAD\_GET\_CONTEXT  THREAD\_SET\_CONTEXT  THREAD\_QUERY\_INFORMATION  THREAD\_SET\_INFORMATION THREAD\_SET\_THREAD\_TOKEN THREAD\_IMPERSONATE THREAD\_DIRECT\_IMPERSONATION THREAD\_SET\_LIMITED\_INFORMATION THREAD\_QUERY\_LIMITED\_INFORMATION |
| Windows OS Process job objects | Windows OS Process Manager (ps.lib) | Process job name | JOB\_OBJECT\_ASSIGN\_PROCESS JOB\_OBJECT\_SET\_ATTRIBUTES JOB\_OBJECT\_QUERY JOB\_OBJECT\_TERMINATE JOB\_OBJECT\_SET\_SECURITY\_ATTRIBUTES |
| Windows OS Registry ConfigJournal objects | Windows OS Registry Configuration Manager (config.lib) | ConfigJournal name | JOURNAL\_RESET\_ACCESS JOURNAL\_NOTIFY\_ACCESS |
| Windows OS Registry Key objects | Windows OS Registry Configuration Manager (config.lib) | Key name | KEY\_QUERY\_VALUE KEY\_SET\_VALUE KEY\_CREATE\_SUB\_KEY KEY\_ENUMERATE\_SUB\_KEYS KEY\_NOTIFY KEY\_CREATE\_LINK |
| (Windows OS) Access token objects | Windows OS security reference monitor (SRM) (se.lib) | Process or Thread ID | TOKEN\_ASSIGN\_PRIMARY TOKEN\_DUPLICATE TOKEN\_IMPERSONATE TOKEN\_QUERY TOKEN\_QUERY\_SOURCE TOKEN\_ADJUST\_PRIVILEGES TOKEN\_ADJUST\_GROUPS TOKEN\_ADJUST\_DEFAULT TOKEN\_ADJUST\_SESSIONID |
| Windows OS Transaction manager objects | Windows OS Transaction management (tm.lib) | Transaction manager name or GUID | TRANSACTIONMANAGER\_QUERY\_INFORMATION TRANSACTIONMANAGER\_SET\_INFORMATION TRANSACTIONMANAGER\_RECOVER TRANSACTIONMANAGER\_RENAME TRANSACTIONMANAGER\_CREATE\_RM |
| Windows OS Transaction objects | Windows OS Transaction management (tm.lib) | Transaction name or GUID | TRANSACTION\_QUERY\_INFORMATION TRANSACTION\_SET\_INFORMATION TRANSACTION\_ENLIST TRANSACTION\_COMMIT TRANSACTION\_ROLLBACK TRANSACTION\_PROPAGATE |
| Windows OS Transaction resource manager objects | Windows OS Transaction management (tm.lib) | Transaction resource manager name or GUID | RESOURCEMANAGER\_QUERY\_INFORMATION RESOURCEMANAGER\_SET\_INFORMATION RESOURCEMANAGER\_RECOVER RESOURCEMANAGER\_ENLIST RESOURCEMANAGER\_GET\_NOTIFICATION RESOURCEMANAGER\_REGISTER\_PROTOCOL RESOURCEMANAGER\_COMPLETE\_PROPAGATION |
| Windows OS Transaction resource manager enlistment objects | Windows OS Transaction management (tm.lib) | Transaction resource manager enlistment name or GUID | ENLISTMENT\_QUERY\_INFORMATION ENLISTMENT\_SET\_INFORMATION ENLISTMENT\_RECOVER ENLISTMENT\_SUBORDINATE\_RIGHTS ENLISTMENT\_SUPERIOR\_RIGHTS |
| Windows OS asynchronous local (inter) process communication manager port objects | Windows OS asynchronous local (inter) process communication manager (alpc.lib) | Connection port name | PORT\_CONNECT |
| Windows OS executive manager callback object | Windows OS executive manager (ex.lib) | Callback names of  "\Callback\EnlightenmentState"; "\Callback\SetSystemTime"; "\Callback\SetSystemState"; "\Callback\PowerState"; "\Callback\ProcessorAdd"; "\Callback\Phase1InitComplete" | CALLBACK\_MODIFY\_STATE |
| Windows OS executive manager event object | Windows OS executive manager (ex.lib) | Event name | EVENT\_QUERY\_STATE EVENT\_MODIFY\_STATE |
| Windows OS executive manager keyed event object | Windows OS executive manager (ex.lib) | Key event name | KEYEDEVENT\_WAIT KEYEDEVENT\_WAKE |
| Windows OS executive manager event pair object | Windows OS executive manager (ex.lib) | Event pair name | no additional object-specific access |
| Windows OS executive manager mutant object | Windows OS executive manager (ex.lib) | Mutant name | MUTANT\_QUERY\_STATE |
| Windows OS executive manager profile object | Windows OS executive manager (ex.lib) | The ID of the process being profiled | PROFILE\_CONTROL |
| Windows OS executive manager semaphore object | Windows OS executive manager (ex.lib) | Semaphore name | SEMAPHORE\_QUERY\_STATE SEMAPHORE\_MODIFY\_STATE |
| Windows OS executive manager timer object | Windows OS executive manager (ex.lib) | Timer name | TIMER\_QUERY\_STATE TIMER\_MODIFY\_STATE |
| Windows OS executive manager thread pool worker factory object | Windows OS executive manager (ex.lib) | Thread pool worker factory name | WORKER\_FACTORY\_QUERY\_INFORMATION WORKER\_FACTORY\_SET\_INFORMATION WORKER\_FACTORY\_RELEASE WORKER\_FACTORY\_WAIT |
| Windows OS Memory manager section objects | Windows OS Memory Manager (mm.lib) | Memory section name | SECTION\_QUERY SECTION\_MAP\_READ SECTION\_MAP\_WRITE SECTION\_MAP\_EXECUTE |
| Windows OS window terminal session objects | Windows OS Memory Manager (mm.lib) | Windows OS window terminal session ID | SESSION\_QUERY\_ACCESS SESSION\_MODIFY\_ACCESS |
| Windows OS object manager kernel object directory objects | Windows OS Object Manager (ob.lib) | Kernel object directory name | DIRECTORY\_QUERY DIRECTORY\_TRAVERSE DIRECTORY\_CREATE\_OBJECT DIRECTORY\_CREATE\_SUBDIRECTORY |
| Windows OS object manager SymbolicLink objects | Windows OS Object Manager (ob.lib) | SymbolicLink object name | SYMBOLIC\_LINK\_QUERY |
| Windows OS debug objects | Windows OS kernel mode debug manager (dbgk.lib) | Debug object name | DEBUG\_READ\_EVENT DEBUG\_PROCESS\_ASSIGN DEBUG\_SET\_INFORMATION DEBUG\_QUERY\_INFORMATION |
| Windows OS file system filter management filter server connection port object | Windows OS file system filter management filter (fltMgr.sys) | Filter server connection port name | FLT\_PORT\_CONNECT |
| Windows OS file system filter management filter client communication port object | Windows OS file system filter management filter (fltMgr.sys) | Filter client communication port name | FLT\_PORT\_CONNECT |
| Windows OS windowstation objects | Windows OS Window Manager (win32k.sys) | Windowstation name | WINSTA\_ENUMDESKTOPS WINSTA\_READATTRIBUTES WINSTA\_ACCESSCLIPBOARD WINSTA\_CREATEDESKTOP WINSTA\_WRITEATTRIBUTES WINSTA\_ACCESSGLOBALATOMS WINSTA\_EXITWINDOWS WINSTA\_ENUMERATE WINSTA\_READSCREEN |
| Windows OS desktop objects | Windows OS Window Manager (win32k.sys) | Desktop name | DESKTOP\_READOBJECTS DESKTOP\_CREATEWINDOW DESKTOP\_CREATEMENU DESKTOP\_HOOKCONTROL DESKTOP\_JOURNALRECORD DESKTOP\_JOURNALPLAYBACK DESKTOP\_ENUMERATE DESKTOP\_WRITEOBJECTS DESKTOP\_SWITCHDESKTOP |
| NTFS file directory objects | NTFS File System (ntfs.sys) | NTFS file directory name | FILE\_LIST\_DIRECTORY FILE\_ADD\_FILE FILE\_ADD\_SUBDIRECTORY FILE\_READ\_EA FILE\_WRITE\_EA FILE\_TRAVERSE FILE\_DELETE\_CHILD FILE\_READ\_ATTRIBUTES FILE\_WRITE\_ATTRIBUTES |
| NTFS file objects | NTFS File System (ntfs.sys) | NTFS file name | FILE\_READ\_DATA FILE\_WRITE\_DATA FILE\_APPEND\_DATA FILE\_READ\_EA FILE\_WRITE\_EA FILE\_EXECUTE FILE\_READ\_ATTRIBUTES FILE\_WRITE\_ATTRIBUTES |
| Windows OS namedpipe objects | Namedpipe File System (npfs.sys) | Namedpipe name | FILE\_READ\_DATA FILE\_WRITE\_DATA FILE\_CREATE\_PIPE\_INSTANCE FILE\_READ\_ATTRIBUTES FILE\_WRITE\_ATTRIBUTES |
| Windows OS mailslot objects | Mailslot File System (msfs.sys) | Mailslot name | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Http application pool object | Windows OS Kernel mode http handler (http.sys) | Application pool name | FILE\_READ\_DATA FILE\_WRITE\_DATA FILE\_APPEND\_DATA FILE\_READ\_EA FILE\_WRITE\_EA FILE\_EXECUTE FILE\_READ\_ATTRIBUTES FILE\_WRITE\_ATTRIBUTES |
| Http URL Reservation object | Windows OS Kernel mode http handler (http.sys) | URL name in the reservation store | HTTP\_ALLOW\_REGISTER\_URL HTTP\_ALLOW\_DELEGATE\_URL |

Table 2: Windows OS user mode named objects and their specific access control elements

|  |  |  |  |
| --- | --- | --- | --- |
| **Named object** | **Resource manager** | **How it is referenced** | **Supported object specific access control elements** |
| Active Directory objects and attributes | Active Directory (ntdsa.dll and ntdsai.dll) | Directory distinguished name | RIGHT\_DS\_CREATE\_CHILD RIGHT\_DS\_DELETE\_CHILD RIGHT\_DS\_DELETE\_SELF RIGHT\_DS\_LIST\_CONTENTS RIGHT\_DS\_WRITE\_PROPERTY\_EXTENDED RIGHT\_DS\_READ\_PROPERTY RIGHT\_DS\_WRITE\_PROPERTY RIGHT\_DS\_DELETE\_TREE RIGHT\_DS\_LIST\_OBJECT RIGHT\_DS\_CONTROL\_ACCESS |
| Windows OS Local Security Account Management (SAM) Local Group objects, not accessible through AD | Windows OS Local Security Account Management (SAM) (samsrv.dll) | Local Group SID | GROUP\_READ\_INFORMATION GROUP\_WRITE\_ACCOUNT GROUP\_ADD\_MEMBER GROUP\_REMOVE\_MEMBER GROUP\_LIST\_MEMBERS |
| Windows OS Local Security Account Management (SAM) Local Alias objects, not accessible through AD | Windows OS Local Security Account Management (SAM) (samsrv.dll) | Local Alias SID | ALIAS\_ADD\_MEMBER ALIAS\_REMOVE\_MEMBER ALIAS\_LIST\_MEMBERS ALIAS\_READ\_INFORMATION ALIAS\_WRITE\_ACCOUNT |
| Windows OS Local Security Account Management (SAM) Local User Account objects, not accessible through AD | Windows OS Local Security Account Management (SAM) (samsrv.dll) | Local User Account SID | USER\_READ\_GENERAL USER\_READ\_PREFERENCES USER\_WRITE\_PREFERENCES USER\_READ\_LOGON USER\_READ\_ACCOUNT USER\_WRITE\_ACCOUNT USER\_CHANGE\_PASSWORD USER\_FORCE\_PASSWORD\_CHANGE USER\_LIST\_GROUPS USER\_READ\_GROUP\_INFORMATION USER\_WRITE\_GROUP\_INFORMATION |
| LSA policy TrustedDomain objects | Windows OS Authentication Service (lsass.exe) | LSA policy TrustedDomain SID | TRUSTED\_QUERY\_DOMAIN\_NAME TRUSTED\_QUERY\_CONTROLLERS TRUSTED\_SET\_CONTROLLERS TRUSTED\_QUERY\_POSIX TRUSTED\_SET\_POSIX TRUSTED\_SET\_AUTH TRUSTED\_QUERY\_AUTH |
| LSA policy Account objects | Windows OS Authentication Service (lsass.exe) | LSA policy Account SID | ACCOUNT\_VIEW ACCOUNT\_ADJUST\_PRIVILEGES ACCOUNT\_ADJUST\_QUOTAS ACCOUNT\_ADJUST\_SYSTEM\_ACCESS |
| LSA policy Secret objects | Windows OS Authentication Service (lsass.exe) | LSA policy Secret name | SECRET\_SET\_VALUE SECRET\_QUERY\_VALUE |
| Windows OS Service objects | Service Control Manager (services.exe) | Windows OS service name | SERVICE\_QUERY\_CONFIG SERVICE\_CHANGE\_CONFIG SERVICE\_QUERY\_STATUS SERVICE\_ENUMERATE\_DEPENDENTS SERVICE\_START SERVICE\_STOP SERVICE\_PAUSE\_CONTINUE SERVICE\_INTERROGATE SERVICE\_USER\_DEFINED\_CONTROL |
| Printer objects | Windows OS print spooler (spoolss.dll with localspl.dll) | Printer name | PRINTER\_ACCESS\_ADMINISTER PRINTER\_ACCESS\_USE |
| Windows OS window management infrastructure (WMI) name space objects | Windows OS window management infrastructure (WMI) (wbemcore.dll) | WMI name space name | WBEM\_ENABLE WBEM\_METHOD\_EXECUTE WBEM\_FULL\_WRITE\_REP WBEM\_PARTIAL\_WRITE\_REP WBEM\_WRITE\_PROVIDER WBEM\_REMOTE\_ACCESS WBEM\_RIGHT\_SUBSCRIBE WBEM\_RIGHT\_PUBLISH |
| Background Intelligent Transfer Service job objects | Background Intelligent Transfer Service (qmgr.dll) | Background Intelligent Transfer Service job ID | BG\_JOB\_QUERY\_PROP BG\_JOB\_SET\_PROP |
| Windows OS Task Schedule Service task objects | Windows OS Task Schedule Service (SchedSvc.dll) | Scheduled task name or GUID | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Event Service log objects | Windows OS Event Service (wevtsvc.dll) | Windows OS Event Service log channel name | EVT\_READ\_ACCESS EVT\_WRITE\_ACCESS EVT\_CLEAR\_ACCESS |
| Windows OS DCOM application objects | Distributed COM Services (rpcss.dll) | Windows OS DCOM application GUID | COM\_RIGHTS\_EXECUTE COM\_RIGHTS\_EXECUTE\_LOCAL COM\_RIGHTS\_EXECUTE\_REMOTE COM\_RIGHTS\_ACTIVATE\_LOCAL COM\_RIGHTS\_ACTIVATE\_REMOTE |
| Windows OS Removable Storage Management Service managed objects of specific supported types | Windows OS Removable Storage Management Service (ntmssvc.dll) | Windows OS Removable Storage Management Service managed object GUID | NTMS\_USE\_ACCESS NTMS\_MODIFY\_ACCESS NTMS\_CONTROL\_ACCESS |
| Windows OS Message Queue objects | Windows OS Message Queue manager service (mqqm.dll) | Windows OS Message Queue format name | MQSEC\_DELETE\_MESSAGE MQSEC\_PEEK\_MESSAGE MQSEC\_WRITE\_MESSAGE MQSEC\_DELETE\_JOURNAL\_MESSAGE MQSEC\_SET\_QUEUE\_PROPERTIES MQSEC\_GET\_QUEUE\_PROPERTIES |
| Windows OS IIS metabase property objects | Windows OS IIS metabase engine service (coadmin.dll) | Windows OS IIS metabase property path | MD\_ACR\_READ MD\_ACR\_WRITE MD\_ACR\_RESTRICTED\_WRITE MD\_ACR\_UNSECURE\_PROPS\_READ MD\_ACR\_ENUM\_KEYS MD\_ACR\_WRITE\_DAC |
| Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Privider objects | Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Service (bfe.dll) | WFP BFE Provider name or GUID (eg. "WFP Built-in IKEEXT Provider") | FWPM\_ACTRL\_ADD FWPM\_ACTRL\_ADD\_LINK FWPM\_ACTRL\_BEGIN\_READ\_TXN FWPM\_ACTRL\_BEGIN\_WRITE\_TXN FWPM\_ACTRL\_CLASSIFY FWPM\_ACTRL\_ENUM FWPM\_ACTRL\_OPEN FWPM\_ACTRL\_READ FWPM\_ACTRL\_READ\_STATS FWPM\_ACTRL\_SUBSCRIBE FWPM\_ACTRL\_WRITE |
| Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Privider context objects | Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Service (bfe.dll) | WFP BFE Provider context name or GUID (eg. "Default Secure Socket AuthIP Policy Provider Context") | FWPM\_ACTRL\_ADD FWPM\_ACTRL\_ADD\_LINK FWPM\_ACTRL\_BEGIN\_READ\_TXN FWPM\_ACTRL\_BEGIN\_WRITE\_TXN FWPM\_ACTRL\_CLASSIFY FWPM\_ACTRL\_ENUM FWPM\_ACTRL\_OPEN FWPM\_ACTRL\_READ FWPM\_ACTRL\_READ\_STATS FWPM\_ACTRL\_SUBSCRIBE FWPM\_ACTRL\_WRITE |
| Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Layer objects | Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Service (bfe.dll) | WFP BFE Layer name or GUID (eg. "Inbound IP Packet v6 Layer") | FWPM\_ACTRL\_ADD FWPM\_ACTRL\_ADD\_LINK FWPM\_ACTRL\_BEGIN\_READ\_TXN FWPM\_ACTRL\_BEGIN\_WRITE\_TXN FWPM\_ACTRL\_CLASSIFY FWPM\_ACTRL\_ENUM FWPM\_ACTRL\_OPEN FWPM\_ACTRL\_READ FWPM\_ACTRL\_READ\_STATS FWPM\_ACTRL\_SUBSCRIBE FWPM\_ACTRL\_WRITE |
| Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Sublayer objects | Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Service (bfe.dll) | WFP BFE Sublayer name or GUID (eg. "WFP Built-in IPsec Tunnel Sublayer") | FWPM\_ACTRL\_ADD FWPM\_ACTRL\_ADD\_LINK FWPM\_ACTRL\_BEGIN\_READ\_TXN FWPM\_ACTRL\_BEGIN\_WRITE\_TXN FWPM\_ACTRL\_CLASSIFY FWPM\_ACTRL\_ENUM FWPM\_ACTRL\_OPEN FWPM\_ACTRL\_READ FWPM\_ACTRL\_READ\_STATS FWPM\_ACTRL\_SUBSCRIBE FWPM\_ACTRL\_WRITE |
| Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Callout objects | Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Service (bfe.dll) | WFP BFE Callout name or GUID (eg. "WFP Built-in IPsec Inbound Transport v6 Layer Callout") | FWPM\_ACTRL\_ADD FWPM\_ACTRL\_ADD\_LINK FWPM\_ACTRL\_BEGIN\_READ\_TXN FWPM\_ACTRL\_BEGIN\_WRITE\_TXN FWPM\_ACTRL\_CLASSIFY FWPM\_ACTRL\_ENUM FWPM\_ACTRL\_OPEN FWPM\_ACTRL\_READ FWPM\_ACTRL\_READ\_STATS FWPM\_ACTRL\_SUBSCRIBE FWPM\_ACTRL\_WRITE |
| Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Filter objects | Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Service (bfe.dll) | WFP BFE Filter name or GUID (eg. "WFP Built-in IKE Exemption Filter") | FWPM\_ACTRL\_ADD FWPM\_ACTRL\_ADD\_LINK FWPM\_ACTRL\_BEGIN\_READ\_TXN FWPM\_ACTRL\_BEGIN\_WRITE\_TXN FWPM\_ACTRL\_CLASSIFY FWPM\_ACTRL\_ENUM FWPM\_ACTRL\_OPEN FWPM\_ACTRL\_READ FWPM\_ACTRL\_READ\_STATS FWPM\_ACTRL\_SUBSCRIBE FWPM\_ACTRL\_WRITE |
| Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Container objects | Windows Filtering Platform (WFP) Base Firewall Engine (BFE) Service (bfe.dll) | WFP BFE Container name or GUID (eg. "WFP Built-in IKE Exemption Filter") | FWPM\_ACTRL\_ADD FWPM\_ACTRL\_ADD\_LINK FWPM\_ACTRL\_BEGIN\_READ\_TXN FWPM\_ACTRL\_BEGIN\_WRITE\_TXN FWPM\_ACTRL\_CLASSIFY FWPM\_ACTRL\_ENUM FWPM\_ACTRL\_OPEN FWPM\_ACTRL\_READ FWPM\_ACTRL\_READ\_STATS FWPM\_ACTRL\_SUBSCRIBE FWPM\_ACTRL\_WRITE |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_permit\_list object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_permit\_list object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_deny\_list object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_deny\_list object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_ac\_enabled object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_ac\_enabled object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_bc\_scan\_enabled object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_bc\_scan\_enabled object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_bss\_type object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_bss\_type object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_show\_denied object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_show\_denied object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_interface\_properties object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_interface\_properties object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_ihv\_control object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_ihv\_control object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_all\_user\_profiles\_order object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_all\_user\_profiles\_order object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_add\_new\_all\_user\_profiles object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_add\_new\_all\_user\_profiles object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_add\_new\_per\_user\_profiles object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_add\_new\_per\_user\_profiles object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_media\_streaming\_mode\_enabled object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_media\_streaming\_mode\_enabled object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |
| Windows OS Wireless LAN (WLAN) AutoConfig service wlan\_secure\_current\_operation\_mode object | Windows OS Wireless LAN (WLAN) AutoConfig service (wlansvc.dll) | The wlan\_secure\_current\_operation\_mode object value | FILE\_READ\_DATA FILE\_EXECUTE FILE\_WRITE\_DATA |

## Addressing 2.1.1.2 “The OS shall associate Discretionary Access Control security attributes with each subject and named object”

The [discretionary access control list (DACL)](http://msdn2.microsoft.com/en-us/library/aa446597(VS.85).aspx) within a [security descriptor structure](http://msdn2.microsoft.com/en-us/library/aa379561(VS.85).aspx) is the mechanism that the Windows OS uses for associating Discretionary Access Control (DAC) security attributes with a named object. This association is set up by the corresponding Windows OS resource manager when the named object is created. The [security descriptor](http://msdn2.microsoft.com/en-us/library/aa379561(VS.85).aspx), including a DACL, may be supplied by the object creator subject, who requests the responsible Windows OS resource manager to create a specific named object.

### Assignment of DACL to named objects which are not Active Directory objects with attributes

The DACL assignment is conducted in the following order:

1. The object's DACL is the DACL from the [security descriptor](http://msdn2.microsoft.com/en-us/library/aa379561(VS.85).aspx) specified by the object's creator subject.
2. The DACL support library functions from ntdll.dll, used by the kernel mode or user mode resource manager, merge any inheritable [Access control Entries (ACEs)](http://msdn2.microsoft.com/en-us/library/aa374912(vs.85).aspx) (if they exist) into the specified DACL unless the [SE\_DACL\_PROTECTED](http://msdn.microsoft.com/en-us/library/ms677634(VS.85).aspx) bit is set in the security descriptor's control bits by the object's creator subject.
3. If the object's creator subject does not specify a [security descriptor](http://msdn2.microsoft.com/en-us/library/aa379561(VS.85).aspx), the kernel mode or user mode resource manager uses the DACL support library functions from ntdll.dll to derive the object's DACL from inheritable ACEs (if they exist).
4. If no security descriptor is specified and there are no inheritable ACEs, the object's DACL is the [default DACL](http://msdn.microsoft.com/en-us/library/aa379623(VS.85).aspx) from the primary or impersonation access token of the object's creator subject. By default, the [default DACL](http://msdn.microsoft.com/en-us/library/aa379623(VS.85).aspx) in the access token allows only the object's creator subject and the local system to obtain access to the object.
5. If there is no specified, inherited, or [default DACL](http://msdn.microsoft.com/en-us/library/aa379623(VS.85).aspx), the kernel mode or user mode resource manager creates the object with no DACL, which allows everyone full access to the object.

### Assignment of DACL to Active Directory objects with attributes

The DACL assignment is conducted in the following order:

1. The Active Directory (AD) object's DACL is the DACL from the object’s security descriptor attribute ([nTSecurityDescriptor](http://msdn2.microsoft.com/en-us/library/ms679006.aspx)) specified by the object's creator subject.
2. The DACL support library functions from ntdll.dll, used by the Windows OS Active Directory, merge any inheritable [Access control Entries (ACEs)](http://msdn2.microsoft.com/en-us/library/aa374912(vs.85).aspx) from the object’s parent AD object into the specified DACL unless the [SE\_DACL\_PROTECTED](http://msdn.microsoft.com/en-us/library/ms677634(VS.85).aspx) bit is set in the control bits of the object’s security descriptor attribute by the object's creator subject.
3. If the object's creator subject does not specify the object’s security descriptor attribute, the Windows OS Active Directory uses the DACL support library functions from ntdll.dll to merge any inheritable ACEs the object’s parent AD object into the default DACL from the [classSchema](http://msdn.microsoft.com/en-us/library/ms680982(VS.85).aspx) AD object for the object's class.
4. If the [classSchema](http://msdn.microsoft.com/en-us/library/ms680982(VS.85).aspx) AD object for the object's class does not have a default DACL, the object's DACL is the [default DACL](http://msdn.microsoft.com/en-us/library/aa379623(VS.85).aspx) from the primary or impersonation access token of the object's creator subject.
5. If there is no specified, inherited, or default DACL, the Windows OS Active Directory creates the object with no DACL, which allows everyone full access to the object.

### Content of an ACE

Each [ACE](http://msdn.microsoft.com/en-us/library/aa374912(vs.85).aspx) specifies

* an ACE type;
* a security identifier (SID) identifying an individual user account or a group (i.e. a trustee);
* an access mask containing applicable standard and object specific access control elements.

Each ACE also has inheritance attributes, associated with it, to specify if the ACE applies

* to the associated object only;
* to its children objects only;
* to both the associated object and its children objects.

In the context of the DAC policy enforcement of the Commercial Grade OS Requirement Set, four types of ACE are relevant, and they are explained as follows.

* [ACCESS\_ALLOWED\_ACE](http://msdn.microsoft.com/en-us/library/aa374847(VS.85).aspx)
  + It is used to grant certain standard and object specific access control elements for an individual user account or a group identified by the SID (i.e. the trustee) in the ACE to an object.
* [ACCESS\_DENIED\_ACE](http://msdn.microsoft.com/en-us/library/aa374879(VS.85).aspx)
  + It is used to deny certain standard and object specific access control elements for an individual user account or a group identified by the SID (i.e. the trustee) field in the ACE to an object.
* [ACCESS\_ALLOWED\_OBJECT\_ACE](http://msdn.microsoft.com/en-us/library/aa374857(VS.85).aspx)
  + It is used to grant certain standard and object specific access control elements for an individual user account or a group identified by the SID (i.e. the trustee) field in the ACE to an AD object, or to a property (i.e. attribute) set of an AD object, or to a property (i.e. attribute) of an AD object, where an AD object is a structure of 3-level hierarchy consisting of the AD object reference, property sets and properties as depicted in the following diagram.



* + It is also used to grant the trustee's right to create a specific class of child objects, where the child and parent object relationship is defined in the AD class schema.
  + It is also used to identify a specific class of child objects that can inherit this ACE, where the child and parent object relationship is defined in the AD class schema.
* [ACCESS\_DENIED\_OBJECT\_ACE](http://msdn.microsoft.com/en-us/library/aa374887(VS.85).aspx)
  + It is used to deny certain standard and object specific access control elements for an individual user account or a group identified by the SID (i.e. the trustee) field in the ACE to an AD object, or to a property (i.e. attribute) set of an AD object, or to a property (i.e. attribute) of an AD object, where an AD object is a structure of 3-level hierarchy consisting of the AD object reference, property sets and properties as depicted in the above diagram.
  + It is also used to deny the trustee's right to create a specific class of child objects, where the child and parent object relationship is defined in the AD class schema.
  + It is also used to identify a specific class of child objects that can inherit this ACE, where the child and parent object relationship is defined in the AD class schema.

### Identification of the owner of an object

The [security descriptor](http://msdn2.microsoft.com/en-us/library/aa379561(VS.85).aspx) associated with a named object contains the “[owner SID](http://msdn.microsoft.com/en-us/library/aa379561(VS.85).aspx)” field which identifies an individual user account or a group as the owner of the object.

Consequently, this Commercial Grade OS Requirement Set “2.1.1.2” requirement is satisfied.

## Addressing 2.1.1.3 “The OS shall enforce the Discretionary Access Control policy on named objects based on the specific types of the subject and object security attributes”

The Commercial Grade OS Requirement Set requires the following types of subject and object security attributes:

1. The authorized user identity and group membership(s) associated with a subject;
2. The {identity, access operations} pairs associated with a named object.

As stated in the “Content of an ACE” section of this paper, the above two types of subject and object security attributes are captured in the description of one of the four ACE types:

* ACCESS\_ALLOWED\_ACE;
* ACCESS\_DENIED\_ACE;
* ACCESS\_ALLOWED\_OBJECT\_ACE;
* ACCESS\_DENIED\_OBJECT\_ACE.

Specifically, the SID of an ACE for identifying an individual user account or a group (i.e. a trustee) is the authorized user identity and group membership(s) associated with a subject, as required in a) above.

An ACE is associated with a named object as the ACE is an entry of the DACL of the named object. Each of the above four ACE types clearly defines an {identity, access operations} pair, as required in b) above.

Consequently, this Commercial Grade OS Requirement Set “2.1.1.3” requirement is satisfied.

## Addressing 2.1.1.4 “The OS shall provide restrictive default values for object security attributes that are used to enforce the Discretionary Access Control policy”

As stated in the following two sections on this paper, the restrictive default value is the default DACL from the primary or impersonation access token of the object's creator subject.

* “Assignment of DACL to named objects which are not Active Directory objects with attributes”;
* “Assignment of DACL to Active Directory objects with attributes”.

By default, the default DACL in the access token allows only the object's creator subject and the local system to obtain access to the object.

Consequently, this Commercial Grade OS Requirement Set “2.1.1.4” requirement is satisfied.

## Addressing 2.1.1.5 “The OS shall ensure that only valid values are accepted for an object’s Discretionary Access Control security attributes”

In Tables 1 and 2 above, the applicable object specific access control elements for each type/class of the Windows OS named objects are stated. For each named object type/class, it is the responsible resource manager that defines the object specific access control elements applicable to the named object type/class. The resource manager specifies the object access control elements in the structure known as [GENERIC\_MAPPING](http://msdn2.microsoft.com/en-us/library/aa446633(vs.85).aspx). In particular, the GenericAll element of the GENERIC\_MAPPING structure specifies an access mask defining all possible access control elements applicable to the object type/class. When the resource manager uses the DACL support library functions from ntdll.dll to compute an effective DACL, with the creator-subject-supplied security descriptor as one of the input elements, it also passes in the [GENERIC\_MAPPING](http://msdn2.microsoft.com/en-us/library/aa446633(vs.85).aspx) structure to the DACL support library functions. Specifically, the access mask of every ACE is subject to a match with the GenericAll element of the passed-in [GENERIC\_MAPPING](http://msdn2.microsoft.com/en-us/library/aa446633(vs.85).aspx) structure. This match effectively filters out bits from the ACE access mask which are not defined in the GenericAll element. As a result, the ACE has only valid values (namely the valid access mask) for a named object’s Discretionary Access Control security attributes.

Consequently, this Commercial Grade OS Requirement Set “2.1.1.5” requirement is satisfied.

## Addressing 2.1.1.6 “the Discretionary Access Control policy mechanism shall provide that named objects are protected from unauthorized access according to a specific set of ordered rules”

The Commercial Grade OS Requirement Set requires the following ordered rules.

1. If the requested mode of access is explicitly denied to the requesting user, deny access;
2. If the requested mode of access is permitted to the requesting user, permit access;
3. If the requested mode of access is explicitly denied to every group of which the requesting user is a member, deny access;
4. If the requested mode of access is permitted to any group of which the requesting user is a member, grant access;
5. Else, deny access.

### Extended DAC policy ordered rules for object security inheritance

Because of the “object security inheritance” concept in the Windows OS, this paper aims to actually demonstrate the following ordered rules (which are an extension of the above) instead. Specifically, the extended rules are highlighted in **bold** texts.

1. If the requested mode of access**, explicitly assigned to the named object,** is explicitly denied to the requesting user, deny access;
2. If the requested mode of access**, explicitly assigned to the named object,** is permitted to the requesting user, permit access;
3. If the requested mode of access**, explicitly assigned to the named object,** is explicitly denied to every group of which the requesting user is a member, deny access;
4. If the requested mode of access**, explicitly assigned to the named object,** is permitted to any group of which the requesting user is a member, grant access;
5. **If the named object has a parent and the requested mode of access inherited from the object's parent is explicitly denied to the requesting user, deny access;**
6. **If the named object has a parent and the requested mode of access inherited from the object's parent is permitted to the requesting user, permit access;**
7. **If the named object has a parent and the requested mode of access inherited from the object's parent is explicitly denied to every group of which the requesting user is a member, deny access;**
8. **If the named object has a parent and the requested mode of access inherited from the object's parent is permitted to any group of which the requesting user is a member, grant access;**
9. Else, deny access.

In addition, it is necessary that the Windows OS also supports the following types of subject and object security attributes as an extension of the subject and object security attributes stated in 2.1.1.3 of the Commercial Grade OS Requirement Set:

1. The authorized user identity and group membership(s) associated with a subject;
2. The {identity, access operations} pairs **explicitly** associated with a named object;
3. **The inheritable {identity, access operations} pair associated with a named object’s parent container.**

### Implementation locations of the Windows OS access determination algorithm

The Windows OS access determination algorithm is implemented in two locations:

* the kernel mode security reference monitor (SRM) [se.lib] as part of ntoskrnl.exe;
* the user mode authorization framework library “authz.dll”.

The SRM exports the following interfaces, which may be used by kernel mode and user mode caller subjects:

* [NtAccessCheck()](http://msdn.microsoft.com/en-us/library/aa374815(VS.85).aspx);
* [NtAccessCheckAndAuditAlarm()](http://msdn.microsoft.com/en-us/library/aa374823(VS.85).aspx);
* [NtAccessCheckByType()](http://msdn.microsoft.com/en-us/library/aa374826(VS.85).aspx);
* [NtAccessCheckByTypeAndAuditAlarm()](http://msdn.microsoft.com/en-us/library/aa374831(VS.85).aspx);
* [NtAccessCheckByTypeResultList()](http://msdn.microsoft.com/en-us/library/aa374836(VS.85).aspx);
* [NtAccessCheckByTypeResultListAndAuditAlarm()](http://msdn.microsoft.com/en-us/library/aa374838(VS.85).aspx);
* [NtAccessCheckByTypeResultListAndAuditAlarmByHandle()](http://msdn.microsoft.com/en-us/library/aa374843(VS.85).aspx).

These interfaces encapsulate the Windows OS access determination algorithm.

The user mode authorization framework library exports the following interfaces, which may be used by user mode caller subjects:

* [AuthzAccessCheck()](http://msdn.microsoft.com/en-us/library/aa375788(VS.85).aspx);
* [AuthzCachedAccessCheck()](http://msdn.microsoft.com/en-us/library/aa375803(VS.85).aspx).

Unlike the case of calling [NtAccessCheckXXX()](http://msdn.microsoft.com/en-us/library/aa374815(VS.85).aspx) exported by the kernel mode security reference monitor, the calling of the user mode authorization framework library occurs totally within the process address space of the caller subject, without having to engage into a user mode kernel mode context switching. Basically, the availability of the user mode authorization framework library allows a user mode resource manager (acting as the caller subject of the framework library) to conduct the necessary access check in a higher performance manner. The performance gain is achieved because a user mode kernel mode context switching is not necessary.

### Windows OS access determination algorithm

Assuming that a DACL is present for an object, the following access determination algorithm is used to determine if a specific desired access mask is granted or denied to a supplied access token (which represents the user account requesting the desired access mask).

1. If there is no ACE in the DACL (i.e. the DACL is empty), then the algorithm terminates and the specific desired access mask is denied.
2. Iteratively process each of the ACEs in the order that they appear in the DACL as described below.
   1. If the inheritance attributes associated with the ACE indicate that the ACE is applicable only to children objects of the associated object, then the ACE is skipped.
   2. If the SID in the ACE does not match any SID in the supplied access token, then the ACE is skipped.
   3. Assuming that a SID match is found, and that the access mask in the ACE matches an access control element in the desired access mask,
      1. if the ACE’s type is ACCESS\_DENIED\_ACE, then the algorithm terminates and the specific desired access mask is denied;
      2. if the ACE’s type is ACCESS\_DENIED\_OBJECT\_ACE,
         * where the ACE does not specify a GUID, then the algorithm terminates and the specific desired access mask is denied;
         * where the ACE specifies a GUID, the GUID represents a property set or a property of the associated object, and the property set or the property is required for an access control element in the desired access mask to be granted to, then the algorithm terminates and the specific desired access mask is denied;
      3. if the ACE’s type is ACCESS\_ALLOWED\_ACE, then the overlap of the access mask in the ACE and the desired access mask is granted for the associated object;
      4. if the ACE’s type is ACCESS\_ALLOWED\_OBJECT\_ACE,
         * where the ACE does not specify a GUID, then the overlap of the access mask in the ACE and the desired access mask is granted for the associated object;
         * where the ACE specifies a GUID, the GUID represents a property set or a property of the associated object, and the property set or the property is required for an access control element in the desired access mask to be granted to, then the overlap of the access mask in the ACE and the desired access mask is granted for the property set or the property of the associated object.
3. After the above iterative process for the ordered ACEs in the DACL reaches its end,
   1. if there is still an access control element in the desired access mask which has not been granted, then the specific desired access mask is denied;
   2. if there is not any access control element in the desired access mask which has not been granted, then the access determination algorithm terminates successfully.

### Windows OS placement of inherited ACEs in the DACL of an object

As mentioned earlier, the merging of any inheritable [Access control Entries (ACEs)](http://msdn2.microsoft.com/en-us/library/aa374912(vs.85).aspx) (if they exist) into a specific DACL of an object is handled in the DACL support library functions from ntdll.dll. These inheritable ACEs are placed at the end of the specific DACL to become inherited ACEs, right after the explicit ACEs already specified in the DACL. Inherited ACEs are placed in the order in which they are inherited. ACEs inherited from the child object's parent come first, following by ACEs inherited from the grandparent, and so on up the tree of objects.

### Ordering of ACEs in an object’s DACL

Due to the “[Windows OS placement of inherited ACEs in an object's DACL](#_WOS_placement_of)”, it is easy to see that the effect of the “Windows OS access determination algorithm” is more restrictive than the effect of the “[Extended DAC policy ordered rules for object security inheritance](#_Extended_DAC_policy)”, if the Windows OS resource managers place the explicit and inheritable ACEs into the DACLs of parent and child objects in the so called “[Windows OS DACL preferred ACE order](http://msdn.microsoft.com/en-us/library/aa379298.aspx)”, which is defined as follows.

1. All explicit ACEs are placed in a collection before any inheritable ACE;
2. Within the collection of explicit ACEs, ACEs identifying an individual user account through their SID field are placed before ACEs identifying a group through their SID field;
3. Within the collection of inheritable ACEs, ACEs identifying an individual user account through their SID field are placed before ACEs identifying a group through their SID field;
4. Within each collection of ACEs identifying an individual user account through their SID field; ACCESS\_DENIED\_ACE or ACCESS\_DENIED\_OBJECT\_ACE ACEs are placed before ACCESS\_ALLOWED\_ACE or ACCESS\_ALLOWED\_OBJECT\_ACE ACEs;
5. Within each collection of ACEs identifying a group through their SID field; ACCESS\_DENIED\_ACE or ACCESS\_DENIED\_OBJECT\_ACE ACEs are placed before ACCESS\_ALLOWED\_ACE or ACCESS\_ALLOWED\_OBJECT\_ACE ACEs.

### Windows OS resource managers following the rules of the Windows OS DACL preferred ACE order

Typically, Windows OS resource managers follow the aforementioned rules for the Windows OS DACL preferred ACE order, especially when dealing with specific DACLs for controlling the authorized access to their named objects. However, on occasions, an exception away from the rules for the Windows OS DACL preferred ACE order is warranted.

For example, the following DACL, namely “DACL X”, is necessary to implement a resource manager’s desired security effects of

* Allowing any member of the “Administrators” local group the read and write access regardless of whether the member resides locally or remotely;
* Disallowing a non-administrator requester any access if the requester resides remotely (i.e. belonging the “Network Users” local group;
* Allowing a local requester the read access if the requester belongs to the “Authenticated Users” local group.



Obviously, DACL X is not in the Windows OS DACL preferred ACE order. Therefore, DACL X does not conform to the “[Extended DAC policy ordered rules for object security inheritance](#_Extended_DAC_policy)” nor the Commercial Grade OS Requirement Set’s DAC policy ordered rules. However, it is effective for enforcing the resource manager’s desired security effects stated in the above in an efficient matter.

Theoretically, one may define a group called “Non Administrator Remote Users”, where its memberships consist of members of the “Network Users” local group, but not identified also as a member of the “Administrators” local group. In other words, the “Non Administrator Remote Users” group is constructed from the “Network Users” by removing its members who are also a member of the “Administrators” group. It is easy to see that the following DACL, namely “DACL Y”, provides the equivalent desired security effects of the resource manager.

* Disallowing a non-administrator requester the read and write access if the requester resides remotely (i.e. belonging the “Non Administrator Remote Users” group;
* Allowing any member of the “Administrators” local group the read and write access regardless of whether the member resides locally or remotely;
* Allowing a local requester the read access if the requester belongs to the “Authenticated Users” local group.



Obviously, DACL Y is in the Windows OS DACL preferred ACE order. Therefore, DACL Y conforms to the “[Extended DAC policy ordered rules for object security inheritance](#_Extended_DAC_policy)” and the Commercial Grade OS Requirement Set’s DAC policy ordered rules.

Unfortunately, while the “Administrators” local group and the “Network Users” local group are built-in groups in the Windows OS, the “Non Administrator Remote Users” group is not defined in the Windows OS. When the developer of a Windows OS resource manager needs to define a hard-coded DACL for controlling the access to a built-in named object of the Windows OS resource manager, the developer would not have the “Non Administrator Remote Users” group, but only the “Administrators” local group and the “Network Users” local group, available for defining the hard-coded DACL. Therefore, the developer has a legitimate reason to define DACL X, which is not in the Windows OS DACL preferred ACE order, because the “Non Administrator Remote Users” group is not available for defining DACL Y, even though DACL Y is in the Windows OS DACL preferred ACE order.

In general, given a DACL (say, DACL P) that is not in the Windows OS DACL preferred ACE order, it is always possible to shift a deny ACE (e.g. an ACCESS\_DENIED\_ACE) to the left hand side in the DACL (making it into, say, DACL Q) in the following manner, while keeping the equivalent desired security effects.

Before the shift to the left hand side:

* DACL P = … … {Allow – Group A; access}; {Deny – Group B; access} … …

After the shift to the left hand side:

* DACL Q = … … {Deny – Group B\A; access}; {Allow – Group A; access} … …

In the above, Group B\A is the group that is constructed from Group B by removing its memberships who are also a member of Group A.

It is noted that the permission that is necessary for creating a group is different from the permission that is necessary for assigning a DACL to a named object. By default, the permission for group creation is granted to an authorized administrator. The permission for DACL assignment to a named object is not limited to an authorized administrator. The DACL assignment permission is available to a subject who possesses the [WRITE\_DAC](http://msdn.microsoft.com/en-us/library/aa379607.aspx) standard access control element applicable to a named object. In other words, a subject, who is able to make a DACL assignment to a named object, need not have the group creation permission. Therefore, a subject, who is allowed to assign DACL P in the aforementioned example, may not have the permission to create Group B\A for converting DACL P into DACL Q, in the case where Group A and Group B exist, but Group B\A does not. Since the option to create Group B\A is not available to the subject, converting DACL P into DACL Q is not possible. As a result, DACL P is the only DACL that gives the subject its desired security effects; even though DACL P is not in the Windows OS DACL preferred ACE order.

Let the operating system automatically create Group B\A in addition of Group A and Group B is also problematic. The main problems are

* the combinatorial explosion of needing to automatically create a group compliment Group U\V for every pair of groups: Group U and Group V;
* keeping the memberships of the group compliment Group U\V synchronized semantically with the memberships of Group U and Group V whenever the memberships of Group U or Group V change.

Consequently, there are very little advantages, but problems, for the Windows OS resource managers to semantically convert a DACL, that is not in the Windows OS DACL preferred ACE order, into another DACL, that is in the Windows OS DACL preferred ACE order.

As a result, the Windows OS resource managers do not enforce any policy to ensure that DACLs for their objects must be in the Windows OS DACL preferred ACE order. As long as an authorized user is allowed to set or change a specific DACL, he/she may specify the DACL with an ACE order that meets his/her intent, using the existing groups, which are available to him/her, to make the DACL assignment. It is not necessary that the DACL must be in the Windows OS DACL preferred ACE order. On the other hand, if the authorized user actually specifies a DACL that is in the Windows OS DACL preferred ACE order, then the Windows OS resource managers keep the preferred ACE order intact.

Consequently, the intent of this Commercial Grade OS Requirement Set “2.1.1.6” requirement is addressed, given the above explanation for the DACL conversion.

# Meeting the “Access Control Polices Discretionary Access Control Policy Management Requirements”

In the Commercial Grade OS Requirement Set, there are three individual management requirements under the heading of “Access Control Polices Discretionary Access Control Policy Management Requirements”. They are listed as “2.1.2.n”, where n = 1, 2, and 3.

## Addressing 2.1.2.1 “The OS shall allow only authorized administrators, object owners, and users with the DAC change authorization the ability to change the access permission associated with a named object”

As mentioned earlier, [WRITE\_DAC](http://msdn.microsoft.com/en-us/library/aa379607.aspx) is one of the standard access control elements applicable to every named object. It corresponds to the right to modify the DACL of a named object. In other words, it corresponds to the DAC change authorization defined in the Commercial Grade OS Requirement Set. It is a valid access control element which may appear in the access mask of any ACE of a DACL.

The Windows OS supports the use of an inheritable “creator owner rights” ACE for determining the specific access rights for granting to the creator owner of a named object. The “creator owner rights” ACE is an ACE of the type ACCESS\_ALLOWED\_ACE or ACCESS\_ALLOWED\_OBJECT\_ACE ACE, where its SID field indicates the SECURITY\_CREATOR\_OWNER\_RIGHTS\_RID for identifying its trustee. Unless there is an explicit (inheritable) “creator owner rights” ACE in a DACL, the owner of an object is always granted WRITE\_DAC and READ\_CONTROL. On the other hand, if there are one or more explicit “creator owner rights” ACEs in the DACL, then these explicit “creator owner rights” ACEs are used in the “Windows OS access determination algorithm” to determine if a specific desired access mask is granted or denied to the owner of an object.

If a Windows OS resource manager does not wish to grant WRITE\_DAC to the object creator subject of its object, then it specifies an explicit “creator owner rights” ACE in the DACL of the object, where the “creator owner rights” ACE does not include WRITE\_DAC in its access mask.

Typically, Windows OS resource managers grant WRITE\_DAC to authorized administrators by introducing WRITE\_DAC into the access mask for inclusion in an ACCESS\_ALLOWED\_ACE or ACCESS\_ALLOWED\_OBJECT\_ACE ACE, which identifies the “Administrators” local group.

Consequently, this Commercial Grade OS Requirement Set “2.1.2.1” requirement is satisfied.

## Addressing 2.1.2.2 “The OS shall allow only authorized administrators the ability to change object ownership”

As mentioned earlier, WRITE\_OWNER is one of the standard access control elements applicable to every named object. It corresponds to the right to change the “owner SID” field of the security descriptor associated with a named object. In other words, it corresponds to the object ownership change ability defined in the Commercial Grade OS Requirement Set. It is a valid access control element which may appear in the access mask desired by a requester subject for a specific object.

In the case of the user mode authorization framework library (authz.dll), if the access mask desired by the requester subject includes WRITE\_OWNER and the “Windows OS access determination algorithm” does not grant WRITE\_OWNER, then the requester subject must have the [SeTakeOwnershipPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK).

In the case of the kernel mode security reference monitor (SRM) [se.lib] as part of ntoskrnl.exe, if the access mask desired by the requester subject includes WRITE\_OWNER and the “Windows OS access determination algorithm” does not grant WRITE\_OWNER, then the requester subject must have the [SeTakeOwnershipPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) or the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK).

By default, the [SeTakeOwnershipPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) is enabled for any member of the “Administrators” local group. Also, by default, the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) is not enabled for any member of the “Administrators” local group. As a member of the “Administrators” local group, the member can explicitly enable the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) for himself/herself.

Finally, the Windows OS supports the use of an inheritable “creator owner rights” ACE for determining the specific access rights for granting to the creator owner of a named object. The inheritable “creator owner rights” ACE contains the SECURITY\_CREATOR\_OWNER\_RIGHTS\_RID in its SID field to identify its trustee. This inheritable “creator owner rights” ACE may be used by an administrator to remove the implicit access rights such as WRITE\_DAC or WRITE\_OWNER that otherwise would be granted to the creator owner of a named object. In the “Windows OS access determination algorithm”, if the requester subject is the creator owner and the security descriptor of the named object being requested contains an inheritable “creator owner rights” ACE, then the algorithm includes the inheritable “creator owner rights” ACE for determining the specific access rights (if any) for granting to the requester subject.

Consequently, this Commercial Grade OS Requirement Set “2.1.2.2” requirement is satisfied.

## Addressing 2.1.2.3 “The OS shall provide only authorized administrators the ability to specify alternative initial values for default Discretionary Access Control object security attributes”

As mentioned earlier, the default DACL from the primary or impersonation access token of the object's creator subject specifies the initial values for default Discretionary Access Control (DAC) object security attributes. This default DACL of an access token may be changed by an authorized subject having the TOKEN\_ADJUST\_DEFAULT access to the access token. The interface for conducting this change is [NtSetInformationToken()](http://msdn.microsoft.com/en-us/library/ms800881.aspx) specifying [TokenDefaultDacl](http://msdn.microsoft.com/en-us/library/aa379591(VS.85).aspx) as the TokenInformationClass value. By default, the TOKEN\_ADJUST\_DEFAULT access is granted to the “Administrators” local group and the local system.

Consequently, this Commercial Grade OS Requirement Set “2.1.2.3” requirement is satisfied.

# Meeting the “Access Control Polices Discretionary Access Control Policy Audit Requirements”

In the Commercial Grade OS Requirement Set, there are three individual audit requirements under the heading of “Access Control Polices Discretionary Access Control Policy Audit Requirements”. They are listed as “2.1.3.n”, where n = 1, 2, and 3.

## Addressing 2.1.3.1 “The OS shall provide the ability to audit all Discretionary Access Control policy decisions”

Similar to DACL, the [system access control list (SACL)](http://msdn.microsoft.com/en-us/library/aa375723(VS.85).aspx) of [SYSTEM\_AUDIT\_ACE](http://msdn.microsoft.com/en-us/library/aa379616(VS.85).aspx) ACEs and [SYSTEM\_AUDIT\_OBJECT\_ACE](http://msdn.microsoft.com/en-us/library/aa379619(VS.85).aspx) ACEs within a [security descriptor structure](http://msdn2.microsoft.com/en-us/library/aa379561(VS.85).aspx) is the mechanism that the Windows OS uses for associating the auditing requirements of the DAC policy decisions with a named object. This association may be set up by the corresponding resource manager if it supports an interface for an authorized subject to do so. If a supporting interface exists, the security descriptor, including a SACL, may be supplied by an authorized subject, who requests the responsible resource manager to associate the auditing requirements of the DAC policy decisions with a specific named object. To associate the auditing requirements, the authorized subject must have the [SeSecurityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK). By default, the [SeSecurityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) is enabled for any member of the “Administrators” local group.

The following are the corresponding security audit event records for auditing the DAC policy decisions:

* [Event ID 4656](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OPEN\_HANDLE\_value) “A handle to an object was requested” for success or failure with the following informational items when available:
  + Security ID:
  + Account Name:
  + Account Domain:
  + Logon ID:
  + Object Server:
  + Object Type:
  + Object Name:
  + Handle ID:
  + Process ID:
  + Process Name:
  + Transaction ID:
  + Accesses:
  + Access Mask:
  + Privileges Used for Access Check:
  + Restricted SID Count:;
* [Event ID 4661](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OPEN\_HANDLE\_OBJECT\_TYPE\_value) “A handle to an object was requested” for success or failure with the following informational items when available:
  + Security ID:
  + Account Name:
  + Account Domain:
  + Logon ID:
  + Object Server:
  + Object Type:
  + Object Name:
  + Handle ID:
  + Process ID:
  + Process Name:
  + Transaction ID:
  + Accesses:
  + Access Mask:
  + Privileges Used for Access Check:
  + Properties:
  + Restricted SID Count:;
* [Event ID 4662](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OBJECT\_OPERATION\_value) “An operation was performed on an object” for success or failure with the following informational items when available:
  + Subject:
  + Security ID:
  + Account Name:
  + Account Domain:
  + Logon ID:
  + Object:
  + Object Server:
  + Object Type:
  + Object Name:
  + Handle ID:
  + Operation:
  + Operation Type:
  + Accesses:
  + Access Mask:
  + Properties:
  + Additional Information:
  + Parameter 1:
  + Parameter 2:;
* [Event ID 4663](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OBJECT\_ACCESS\_value) “An attempt was made to access an object” for success or failure with the following informational items when available:
  + Security ID:
  + Account Name:
  + Account Domain:
  + Logon ID:
  + Object Server:
  + Object Type:
  + Object Name:
  + Handle ID:
  + Process ID:
  + Process Name:
  + Accesses:
  + Access Mask:.

Consequently, this Commercial Grade OS Requirement Set “2.1.3.1” requirement is addressed.

## Addressing 2.1.3.2 “The OS shall provide the ability to audit all changes to an object’s Discretionary Access Control security attributes”

The following security audit event record is generated when the security descriptor of a kernel mode named object listed in Table 1 is changed successfully.

* [Event ID 4670](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_SD\_CHANGE\_value) “Permissions on an object were changed” with the following informational items when available:
  + Subject:
  + Security ID:
  + Account Name:
  + Account Domain:
  + Logon ID:
  + Object:
  + Object Server:
  + Object Type:
  + Object Name:
  + Handle ID:
  + Process:
  + Process ID:
  + Process Name:
  + Permissions Change:
  + Original Security Descriptor:
  + New Security Descriptor:.

For a Windows OS AD object, its security descriptor is stored in the object’s security descriptor attribute ([nTSecurityDescriptor](http://msdn2.microsoft.com/en-us/library/ms679006.aspx)). Therefore, the following security audit event record is generated when the security descriptor of a Windows OS AD object is changed successfully.

* [Event ID 5136](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_DS\_OBJECT\_MODIFY\_value) “A directory service object was modified” with the following informational items when available:
  + Subject:
  + Security ID:
  + Account Name:
  + Account Domain:
  + Logon ID:
  + Directory Service:
  + Name:
  + Type:
  + Object:
  + GUID:
  + Class:
  + Attribute:
  + LDAP Display Name:
  + Syntax (OID):
  + Value:
  + Operation:
  + Type:
  + Correlation ID:
  + Application Correlation ID:.

Unfortunately, the responsible Windows OS resource managers of the other user mode named objects listed in Table 2 do not necessary generate a security descriptor change audit record similar to [Event ID 4670](http://support.microsoft.com/kb/947226) above when the security descriptor associated with one of their named objects is changed. As a result, this Commercial Grade OS Requirement Set “2.1.3.2” requirement is only partially addressed.

## Addressing 2.1.3.3 “The OS shall provide the ability to audit all modifications to default Discretionary Access Control object security attribute values”

This requirement is only partially satisfied.

In the case where the effective ACEs of an object (as the default DAC object security attribute values) are derived from the inheritable ACEs of the object’s parent, the modifications of the inheritable ACEs of the object’s parent are audited. This conclusion is based on the following facts:

* the modification of the security descriptor of the object’s parent is audited because of the Commercial Grade OS Requirement Set “2.1.3.2” requirement;
* the DACL of the security descriptor of the object’s parent includes inheritable ACEs.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “2.1.2.3” requirement, the default DACL from the primary or impersonation access token of an object's creator subject may be changed by an authorized subject having the TOKEN\_ADJUST\_DEFAULT access to the access token. However, there is no generation of an audit record to record the change from the old value of the default DACL to the new value of the default DACL.

# Meeting the “Access Control Polices Mandatory Integrity Control Policy Functional Requirements”

In the Commercial Grade OS Requirement Set, there are six individual functional requirements under the heading of “Access Control Polices Mandatory Integrity Control Policy Functional Requirements”. They are listed as “2.2.1.n”, where n = 1, 2, 3, 4, 5 and 6.

## Addressing 2.2.1.1 “The OS shall enforce a Mandatory Integrity Control policy on untrusted subjects, named objects, and network interfaces and all operations that cause information to flow among them”

Similar to DACL, the [system access control list (SACL)](http://msdn.microsoft.com/en-us/library/aa375723(VS.85).aspx) of [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACEs within a [security descriptor structure](http://msdn2.microsoft.com/en-us/library/aa379561(VS.85).aspx) is the mechanism that the Windows OS uses for associating Mandatory Integrity Control (MIC) security attributes with a named object. The list of the [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACEs within the SACL of a security descriptor is called a mandatory access control list (MACL). This association is set up by the corresponding resource manager when the named object is created. A security descriptor, including a MACL, may be modified by an authorized subject, who possesses the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK), to request the responsible resource manager to update the MIC security attributes of a specific named object.

### Windows OS named objects subject to the MIC policy

The Windows OS kernel mode named object types/classes (listed in Table 1) are subject to the MIC policy. Specific Windows OS user mode named object types/classes (listed in Table 2) also are subject to the MIC policy, depending on their corresponding resource managers. Specifically, a resource managers needs to use the [NtAccessCheckXXX()](http://msdn.microsoft.com/en-us/library/aa374815(VS.85).aspx) interfaces (rather than the [AuthzXXXAccessCheck()](http://msdn.microsoft.com/en-us/library/aa375788(VS.85).aspx) interfaces to ensure the enforcement of the MIC policy. In other words, the [NtAccessCheckXXX()](http://msdn.microsoft.com/en-us/library/aa374815(VS.85).aspx) interfaces conduct the mandatory integrity check while the [AuthzXXXAccessCheck()](http://msdn.microsoft.com/en-us/library/aa375788(VS.85).aspx) interfaces do not. Those Windows OS user mode named object types/classes (listed in Table 2), which are subject to the MIC policy, are the following:

* Windows OS Service objects;
* Printer objects;
* Background Intelligent Transfer Service job objects;
* Windows OS Task Schedule Service task objects;
* Windows OS Event Service log objects;
* Windows OS DCOM application objects;
* Windows OS Removable Storage Management Service managed objects of specific supported types;
* Windows OS Message Queue objects.

### Windows OS Network List Manager (NLM)

In Windows OS, network interfaces are not recognized as a named object directly. However, there is the [Windows OS Network List Manager (NLM)](http://msdn.microsoft.com/en-us/library/aa965303(VS.85).aspx), implemented in netprofm.dll, to identify available networks and return network attribute data to applications or user subjects. The NLM enumerates

* all networks the local machine has ever seen;
* just the connected networks;
* just the disconnected networks.

Its interface also makes it easy for a user subject to enumerate the network interfaces on a machine. The NLM determines the properties of a network connection: ID, name, description, type (managed/authenticated), state (connected, disconnected, IPV4 local access, IPV4 Internet access, IPV6 local access, IPV6 Internet access), and others. In addition, the NLM supports the following network categories for a specific network:

* NLM\_NETWORK\_CATEGORY\_PUBLIC
  + the network is a public (untrusted) network;
* NLM\_NETWORK\_CATEGORY\_PRIVATE
  + the network is a private (trusted) network;
* NLM\_NETWORK\_CATEGORY\_DOMAIN\_AUTHENTICATED
  + the network is authenticated against an Active Directory domain.

**In a separate paper, we would attempt to discuss how the NLM and its associated network infrastructure within the local Windows OS would meet the intents of the authors of the Commercial Grade OS Requirement Set, even though the actual network interfaces do not support the MIC policy.**

As a result, other than the case of network interfaces, this Commercial Grade OS Requirement Set “2.2.1.1” requirement is addressed.

## Addressing 2.2.1.2 “The OS shall associate an integrity label with each subject, named object, and network interface that accurately represents its integrity level”

### Windows OS integrity levels

Five integrity levels are defined for supporting the MIC policy in Windows OS. They are:

* MandatoryLevelUntrusted (0)
  + which is assigned to the “Anonymous Logon Token” for representing an unauthenticated user subject not belonging to the “everyone” (i.e. “World”) group;
* MandatoryLevelLow (1)
  + which is assigned to the “Anonymous Logon Token” for representing an unauthenticated user subject belonging to the “Everyone” (i.e. “World”) group;
* MandatoryLevelMedium (2);
* MandatoryLevelHigh (3);
* MandatoryLevelSystem (4).

### Windows OS algorithm to assign the integrity level to an access token for a Windows OS process

Just like the default DACL value in an access token, the integrity level value is assigned to an access token when the Windows OS Authentication Service requests the Windows OS SRM to create the access token. This integrity level value in the access token therefore contains the integrity level of the subject represented by the access token. The Windows OS Authentication Service uses the following rules in its internal “Integrity Level From User And Group Sids” function to determine the integrity level for an authenticated subject.

* If the authenticated subject is the local system or the local service, or the network service, then the subject’s integrity level is MandatoryLevelSystem (4);
* If the authenticated subject belongs to the “Administrators” group and the subject should be running in the elevated mode (i.e. [TokenElevationTypeFull](http://msdn.microsoft.com/en-us/library/bb530718.aspx)), then the subject’s integrity level is MandatoryLevelHigh (3);
* If the authenticated subject possesses the following privileges and the subject should be running in the elevated mode (i.e. [TokenElevationTypeFull](http://msdn.microsoft.com/en-us/library/bb530718.aspx)), then the subject’s integrity level is MandatoryLevelHigh (3)
  + [SeCreateTokenPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK);
  + [SeTcbPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK);
  + [SeTakeOwnershipPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK);
  + [SeBackupPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK);
  + [SeRestorePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK);
  + [SeDebugPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK);
  + [SeImpersonatePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK);
  + [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK);
  + [SeLoadDriverPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK);
* If the authenticated subject belongs to the “Users” group or the “Everyone” (i.e. “World”) group or the subject should be running in the limited mode (i.e. [TokenElevationTypeLimited](http://msdn.microsoft.com/en-us/library/bb530718.aspx)), then the subject’s integrity level is MandatoryLevelMedium (2);
* Else, the subject’s integrity level is MandatoryLevelUntrusted (0).

When a new Windows OS process is created for a subject due to the subject’s launching of an executable file in the process, the Windows OS process manager has the responsibility to duplicate an access token from that of the subject. This duplicated access token would then be assigned to the newly created Windows OS process for the subject in its internal “Initialize Process Security” function. In its internal “Initialize Process Security” function, the Windows OS process manager follows the “minimum between the launching subject integrity level and the executable file integrity level” policy (namely [TOKEN\_MANDATORY\_POLICY\_NEW\_PROCESS\_MIN](http://msdn.microsoft.com/en-us/library/bb394728.aspx)) for the newly created process. The “minimum between the launching subject integrity level and the executable file integrity level” policy specifies that the duplicated access token assigned to the newly created process for representing the subject has the integrity level of the lesser of the launching subject’s integrity level and the executable file’s integrity level. This policy helps protect a user subject who launches untrustworthy code from malicious acts performed by that code. Moreover, the user data, which is at the typical integrity level of MandatoryLevelMedium (2), is write-protected against this newly created process of a lower integrity level. For example, if the administrator user subject executes a low (i.e. MandatoryLevelLow (1)) integrity program, the access token for the newly created process operates with the low (i.e. MandatoryLevelLow (1)) integrity level.

For an existing access token, its integrity level can be modified through the [NtSetInformationToken()](http://msdn.microsoft.com/en-us/library/ms800881.aspx) interface using the [TokenIntegrityLevel](http://msdn.microsoft.com/en-us/library/aa379626(VS.85).aspx) TOKEN\_INFORMATION\_CLASS enumeration. Specifically, to increase the integrity level of an access token, the caller subject must have the [SeTcbPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK).

### MIC access operations on named objects under the MIC policy

As mentioned earlier, for each named object class/type, its resource manager specifies the object-specific access control elements in the structure known as [GENERIC\_MAPPING](http://msdn2.microsoft.com/en-us/library/aa446633(vs.85).aspx). In particular,

* the GenericRead element of the [GENERIC\_MAPPING](http://msdn2.microsoft.com/en-us/library/aa446633(vs.85).aspx) structure defines the read-like object-specific access operations;
* the GenericWrite element of the [GENERIC\_MAPPING](http://msdn2.microsoft.com/en-us/library/aa446633(vs.85).aspx) structure defines the write-like object-specific access operations;
* the GenericExecute element of the [GENERIC\_MAPPING](http://msdn2.microsoft.com/en-us/library/aa446633(vs.85).aspx) structure defines the execute-like object-specific access operations.

In Windows OS, three access operation classes are defined to support the MIC policy and they are the following:

* read ([SYSTEM\_MANDATORY\_LABEL\_NO\_READ\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx))
  + it includes all the read-like object-specific access operations and the READ\_CONTROL standard right;
* write ([SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx))
  + it includes all the write-like object-specific access operations and the WRITE\_DAC, ACCESS\_SYSTEM\_SECURITY, WRITE\_OWNER and DELETE standard rights;
* execute ([SYSTEM\_MANDATORY\_LABEL\_NO\_EXECUTE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx))
  + it includes all the execute-like object-specific access operations and the SYNCHRONIZE standard right.

### Content of a SYSTEM\_MANDATORY\_LABEL\_ACE ACE

A [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE contains:

* an integrity level value of
  + MandatoryLevelLow (1);
  + MandatoryLevelMedium (2);
  + MandatoryLevelHigh (3);
  + MandatoryLevelSystem (4)

representing the integrity level of the named object that the [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE applies to;

* one or more of the three access operation classes defined in the above to support the MIC policy.

In the context of MIC policy enforcement, the [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE has the following purposes.

* If the ACE specifies the read ([SYSTEM\_MANDATORY\_LABEL\_NO\_READ\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class, then a subject of integrity level lesser than the integrity level of the ACE is denied any read-like object-specific access operations or the READ\_CONTROL standard right to the named object associated with the ACE.
* If the ACE specifies the write ([SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class, then a subject of integrity level lesser than the integrity level of the ACE is denied any write-like object-specific access operations or the WRITE\_DAC, ACCESS\_SYSTEM\_SECURITY, WRITE\_OWNER or DELETE standard rights to the named object associated with the ACE.
* If the ACE specifies the execute ([SYSTEM\_MANDATORY\_LABEL\_NO\_EXECUTE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class, then a subject of integrity level lesser than the integrity level of the ACE is denied any execute-like object-specific access operations or the SYNCHRONIZE standard right to the named object associated with the ACE.

### Resource Managers assigning SYSTEM\_MANDATORY\_LABEL\_ACE ACEs to named objects

As mentioned earlier, the [system access control list (SACL)](http://msdn.microsoft.com/en-us/library/aa375723(VS.85).aspx) of [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACEs within a [security descriptor structure](http://msdn2.microsoft.com/en-us/library/aa379561(VS.85).aspx) is the mechanism that Windows OS uses for associating Mandatory Integrity Control (MIC) security attributes with a named object. The list of the [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACEs within the SACL of a security descriptor is called a mandatory access control list (MACL). This association is set up by the corresponding resource manager when the named object is created. A security descriptor, including a MACL, may be modified by an authorized subject, who possesses the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK), to request the responsible resource manager to update the MIC security attributes of a specific named object.

Specifically, a resource manager uses the [RtlAddMandatoryAce()](http://msdn.microsoft.com/en-us/library/aa965464(VS.85).aspx) function to assign a [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE into the [security descriptor structure](http://msdn2.microsoft.com/en-us/library/aa379561(VS.85).aspx) associated with a specific named object that supports the MIC policy.

For any named object, subject to the MIC policy, if a [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE is not found in the associated [security descriptor structure](http://msdn2.microsoft.com/en-us/library/aa379561(VS.85).aspx), the MIC enforcement algorithm (residing in the Windows OS kernel mode security reference monitor [se.lib]) uses the following default [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE of:

* an integrity level value of MandatoryLevelMedium (2);
* the write ([SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class.

As a result, other than the case of network interfaces, this Commercial Grade OS Requirement Set “2.2.1.2” requirement is met.

## Addressing 2.2.1.3 “The OS shall enforce the Mandatory Integrity Control policy based on the specific types of the subject and object integrity attributes”

The Commercial Grade OS Requirement Set requires the following types of subject and object integrity attributes:

1. integrity labels of subjects;
2. integrity labels of objects;
3. integrity labels consisting of at least 2 definable hierarchical levels or a set of at least 2 definable non-hierarchical categories;
4. any additional integrity attributes (e.g. privileges).

As mentioned earlier, the Windows OS uses the Windows OS integrity levels for both subjects and named objects for its MIC policy enforcement. Evidently, these Windows OS integrity levels are hierarchical levels and not non-hierarchical categories.

This requirement is addressed by the MIC enforcement algorithm implemented in the Windows OS kernel mode security reference monitor (SRM) [se.lib]. The execution of the MIC enforcement algorithm occurs in the [NtAccessCheckXXX()](http://msdn.microsoft.com/en-us/library/aa374815(VS.85).aspx) interfaces. It enforces the MIC policy by returning [STATUS\_ACCESS\_DENIED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) according to the following MIC policy enforcement rules.

* If the named object’s specific or default [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE specifies the read ([SYSTEM\_MANDATORY\_LABEL\_NO\_READ\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class, then a subject of integrity level lesser than the integrity level of the ACE is denied the requests of any read-like object-specific access operations or the READ\_CONTROL standard right to the named object associated with the ACE.
* If the named object’s specific or default [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE specifies the write ([SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class, then a subject of integrity level lesser than the integrity level of the ACE is denied the requests of any write-like object-specific access operations or the WRITE\_DAC, ACCESS\_SYSTEM\_SECURITY, WRITE\_OWNER or DELETE standard rights to the named object associated with the ACE.
* If the named object’s specific or default [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE specifies the execute ([SYSTEM\_MANDATORY\_LABEL\_NO\_EXECUTE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class, then a subject of integrity level lesser than the integrity level of the ACE is denied the requests of any execute-like object-specific access operations or the SYNCHRONIZE standard right to the named object associated with the ACE.

As a result, other than the case of network interfaces, this Commercial Grade OS Requirement Set “2.2.1.3” requirement is met.

## Addressing 2.2.1.4 “The OS shall enforce specific relationships for any two valid integrity labels”

The Commercial Grade OS Requirement Set requires one of the following three selections of specific relationships for any two valid integrity labels.

1. For hierarchical integrity policy schemes: there exists an ordering function that, given two valid integrity labels, determines if the integrity labels are equal or if one integrity label is greater than the other;
2. For non-hierarchical integrity attributes schemes, integrity labels are composed of a set of distinct and non-comparable integrity attributes;
3. For integrity policy schemes that include both hierarchical and non-hierarchical components:
   1. There exists an ordering function that, given two valid integrity labels, determines if the integrity labels are equal or if one integrity label dominates the other.
   2. There exists a “least upper bound” in the set of integrity labels, such that, given any two valid integrity labels, there is a valid integrity label that dominates the two valid integrity labels.
   3. There exists a “greatest lower bound” in the set of integrity labels, such that, given any two valid integrity labels, there is a valid integrity label that is dominated by the two valid integrity labels.

The Windows OS supports only a hierarchical integrity policy scheme, namely 1) in the above. As mentioned earlier, the Windows OS integrity levels are defined as follows.

* MandatoryLevelUntrusted (0);
* MandatoryLevelLow (1);
* MandatoryLevelMedium (2);
* MandatoryLevelHigh (3);
* MandatoryLevelSystem (4).

As a result, other than the case of network interfaces, this Commercial Grade OS Requirement Set “2.2.1.4” requirement is met.

## Addressing 2.2.1.5 “The OS shall permit an information flow among subjects and objects based on a specific set of rules”

The Commercial Grade OS Requirement Set requires one of the following two selections of specific sets of rules for information flows among subjects and objects.

1. For hierarchical integrity attributes schemes:
   1. If the integrity label of the subject is greater than or equal to the integrity label of the object, then a write (the flow of information from the subject to the object) is permitted;
   2. If the integrity label of the object is greater than or equal to the integrity label of the subject, then a read (the flow of information from the object to the subject) is permitted;
   3. If the information flow is between objects, the integrity label of the source object must be greater than or equal to the integrity label of the destination object.
2. For non-hierarchical integrity attributes schemes:
   1. If the integrity label of the object is equal to the integrity label of the subject, then the flow of information between the object and the subject is permitted;
   2. If the integrity label of the subject dominates the integrity label of the object (i.e. the object’s label is a subset of the subject’s label), then a write (the flow of information from the subject to the object) is permitted;
   3. If the integrity label of the object dominates the integrity label of the subject (i.e. the subject’s label is subset of the object’s label), then a read (the flow of information from the object to the subject) is permitted;
   4. If the integrity label of the subject and the integrity label of the object have no relationship (i.e. one is not a subset of the other) then they are non-comparable and as a consequence information flow is not allowed.

The Windows OS supports only a hierarchical integrity policy scheme, namely 1) in the above, with the five Windows OS integrity levels.

This requirement is addressed by the MIC enforcement algorithm implemented in the Windows OS kernel mode security reference monitor (SRM) [se.lib]. The execution of the MIC enforcement algorithm occurs in the [NtAccessCheckXXX()](http://msdn.microsoft.com/en-us/library/aa374815(VS.85).aspx) interfaces. It enforces the MIC policy by returning [STATUS\_ACCESS\_DENIED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) according to the following MIC policy enforcement rules.

* If the named object’s specific or default [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE specifies the read ([SYSTEM\_MANDATORY\_LABEL\_NO\_READ\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class, then a subject of integrity level lesser than the integrity level of the ACE is denied the requests of any read-like object-specific access operations or the READ\_CONTROL standard right to the named object associated with the ACE.
* If the named object’s specific or default [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE specifies the write ([SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class, then a subject of integrity level lesser than the integrity level of the ACE is denied the requests of any write-like object-specific access operations or the WRITE\_DAC, ACCESS\_SYSTEM\_SECURITY, WRITE\_OWNER or DELETE standard rights to the named object associated with the ACE.
* If the named object’s specific or default [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE specifies the execute ([SYSTEM\_MANDATORY\_LABEL\_NO\_EXECUTE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class, then a subject of integrity level lesser than the integrity level of the ACE is denied the requests of any execute-like object-specific access operations or the SYNCHRONIZE standard right to the named object associated with the ACE.

We note that the MIC policy enforcement rules of the Windows OS do not address the above 1) b) and c) requirements of the Commercial Grade OS Requirement Set hierarchical integrity attributes schemes fully because a subject with a higher Windows OS integrity level certainly can read a named file object of a lower Windows OS integrity level. However it is believed that the above 1) b) and c) requirements are only intentional especially for Windows OS file objects which

* contain arbitrary executable file data;
* have an arbitrary file name;
* reside in an arbitrary file directory location.

For other Windows OS named objects listed in Table 1 and Table 2, the risk of a higher integrity level subject reading data from a lower integrity level object is a lesser security concern because the named objects being read from reside in known locations and have relatively familiar object names. Due to the known locations and relatively familiar object names, the Windows OS [SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) MIC policy enforcement would have already ensured that these named objects being read could not have been written by a lower integrity subject. Furthermore, the higher integrity subjects (typically the local system, the local service and the network service of the Windows OS) expect only certain data structures or data formats when reading the named objects that they need. Therefore, effective countermeasure validations based on the expected data structures or data formats easily occur within the Windows OS processes or threads of higher integrity subjects before acting upon the read data.

On the other hand, the risk of a higher integrity level subject reading data from a lower integrity level named file object for its executable file content(s) is certainly a security concern. The Windows OS approach to mitigate this specific risk is the functionalities of the Windows OS Administrative Privileged Application Launching Service, which is described in the following section of this paper.

* “Appendix A: Administrative Privileged Application Launching Service”.

Additionally, the Windows OS does not implement a mechanism which allows a named object to cause an information flow to another named object without involving a subject who has been granted the necessary access rights in the acquisition of valid object handles to both named objects. Therefore, the above 1) c) requirement of the Commercial Grade OS Requirement Set hierarchical integrity attributes schemes is moot.

As a result, we believe that the collective support, from

* the Windows OS [SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) MIC policy enforcement;

the functionalities of the Windows OS Administrative Privileged Application Launching Service

* Appendix A: Administrative Privileged Application Launching Service,

is capable to sufficiently address the intents behind this Commercial Grade OS Requirement Set “2.2.1.5” requirement.

## Addressing 2.2.1.6 “The OS shall enforce that all definition of integrity labels are unique”

This requirement is moot for the Windows OS, as the Windows OS supports only a hierarchical integrity policy scheme, namely 1) in the above hierarchical integrity attributes schemes, with the five Windows OS integrity levels. In additional, the five Windows OS integrity levels are fixed.

# Meeting the “Access Control Polices Mandatory Integrity Control Policy Management Requirements”

In the Commercial Grade OS Requirement Set, there are two individual management requirements under the heading of “Access Control Polices Mandatory Integrity Policy Management Requirements”. They are listed as “2.2.2.n”, where n = 1 and 2.

## Addressing 2.2.2.1 “The OS shall provide the ability to set the system-wide definition of integrity labels to authorized administrators”

This requirement is moot for Windows OS, as Windows OS supports only a hierarchical integrity policy scheme, with the five Windows OS integrity levels. In additional, the five Windows OS integrity levels are fixed.

## Addressing 2.2.2.2 “The OS shall provide authorized administrators the ability to change integrity labels of subjects and objects”

As mentioned earlier, for an existing access token, its integrity level can be modified through the [NtSetInformationToken()](http://msdn.microsoft.com/en-us/library/ms800881.aspx) interface using the [TokenIntegrityLevel](http://msdn.microsoft.com/en-us/library/aa379626(VS.85).aspx) TOKEN\_INFORMATION\_CLASS enumeration. Specifically, to increase the integrity level of an access token, the caller subject must have the [SeTcbPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK).

Also mentioned earlier, a security descriptor, including a MACL, may be modified by an authorized subject. The authorized subject must possess the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) to request the responsible resource manager to update the MIC security attributes of a specific named object.

this Commercial Grade OS Requirement Set “2.2.2.2” requirement is met.

# Meeting the “Access Control Polices Mandatory Integrity Control Policy Audit Requirements”

In the Commercial Grade OS Requirement Set, there are three individual audit requirements under the heading of “Access Control Polices Mandatory Integrity Control Policy Audit Requirements”. They are listed as “2.2.3.n”, where n = 1, 2, and 3.

## Addressing 2.2.3.1 “The OS shall provide the ability to audit all Mandatory Integrity Control policy decisions”

This requirement is addressed by the Windows OS using its same mechanisms for addressing the Commercial Grade OS Requirement Set “2.1.3.1” requirement.

## Addressing 2.2.3.2 “The OS shall provide the ability to audit the setting and changing of system-wide integrity label definitions”

This requirement is moot for the Windows OS, as the Windows OS supports only a hierarchical integrity policy scheme, with the five Windows OS integrity levels. In additional, the five Windows OS integrity levels are fixed.

## Addressing 2.2.3.3 “The OS shall provide the ability to audit the changing of subject and object integrity labels”

In the case of an authorized modification of the integrity label (i.e. the [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE within the security descriptor) associated with a named object subject to the MIC policy, the corresponding auditing is supported. The Windows OS uses its same mechanisms for addressing the Commercial Grade OS Requirement Set “2.1.3.2” requirement.

In the case of an authorized modification of the integrity label value of an existing (Windows OS) access token, the corresponding auditing is not supported, as this modification is granted only to a subject possessing the [SeTcbPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK). Therefore, this requirement is only partially addressed.

# Meeting the “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 14 individual functional requirements under the heading of “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Functional Requirements”. They are listed as “3.1.1.n”, where n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, and 14.

## Addressing 3.1.1.1 “The OS shall require each user to be uniquely identified and successfully authenticated by means of a password before allowing any actions on behalf of that user”

As highlighted in the “User authentication” overview section earlier when addressing the “1.1.1.1” requirement, user authentication is in action when a user locally or remotely attempts to gain access to protected resources or services through interfaces implemented or supported by the Windows OS. The user is represented by a unique security identifier (ID) (SID) upon the successful user authentication. Every Windows OS process or thread created by the Windows OS process manager is assigned the association with a user SID, which is used for subsequent user identification purpose.

### Local user logging on

In the local user logging on case, a user interacts with the Windows OS physically using a keyboard, a mouse, or another type of input device, which are physically attached to the local machine. If the user has not logged on to the local machine, then s/he needs to specify

* a supported name (e.g. user principal name or NETBIOS domain name of a user account name);
* a selection for a public key certificate

of a user account for attempting to logon to the specified user account. It is the combination of the Windows OS (window terminal) session specific window logon state maintaining service (aka winlogon.exe) and the Windows OS (window terminal) session specific window logon user interface service (aka logonui.exe) to provide an interactive user interface for a user to logon to the specified user account. If the logging on is successfully, the subsequent activities of the logged on user account, in terms of Windows OS processes belonging to the account, reside in a newly created (window terminal) session maintained by the Windows OS local window terminal service (aka lsm.exe). In this winlogon.exe/logonui.exe combination,

* winlogon.exe provides and maintains a secure display area (i.e. a secure desktop) for secure processes/applications, such as password collecting applications (e.g. logonui.exe), to exist in this display area to interact with the user; while
* logonui.exe (residing in the secure display area after being launched by winlogon.exe) actually displays the well-recognized dialog for interacting with the user to collect a supported name of the user’s desired user account to logon and the associated applicable credentials (i.e. authentication data), such as password or smart card PIN, as appropriate.

After winlogon.exe forwards the collected credentials to the Windows OS authentication service (aka lsass.exe), the Windows OS authentication service attempts to authenticate the user to his/her specified user account, using the collected credentials. In the case where the Windows OS authentication service has decided a successful authentication of the user to the specified user account, with the assistance from a Windows OS credential verification specific component[[3]](#footnote-4), the Windows OS authentication service requests the Windows OS security reference monitor (SRM) to create an access token for the logged on user account. The access token includes the user’s unique security ID (SID), which represents the logged on user account. The SID is received by the responsible Windows OS credential verification specific component of the Windows OS authentication service from either

* the local Windows OS Local Security Account Management (SAM)[[4]](#footnote-5), in the case where the user account is a local user account defined in the local Windows OS running on the machine where the logging on occurs;
* a Windows OS domain controller (collocating with a Kerberos key distribution center)[[5]](#footnote-6), in the case where the user account is a domain-wide user account defined in the Windows OS-enabled distributed OS environment, which includes the local machine of the logging on user as a member computer.

Except the local built-in “Guest” user account ([DOMAIN\_USER\_RID\_GUEST](http://msdn.microsoft.com/en-us/library/aa379649.aspx)) defined in the Windows OS running on an individual machine, all the other local or domain-wide user accounts can be configured by an administrator or a domain account operator to have an associated credential in the form of a password, or a smart card, or both. Hence, except the local built-in “Guest” user account, each of the user accounts can be deployed to represent a user uniquely, assuming that the user keeps his/her credential(s) (i.e. password or smart card PIN) secret, and his/her smart card only in his/her possession. The local built-in “Guest” user account is not suitable for representing a user, as the logging on to the “Guest” user account does not require any credential. Unless the Windows OS in question is a “Home” SKU (i.e. Windows Vista Home Premium or Windows Vista Home Basic), the “Guest” user account is disabled, by default, by the local Windows OS Local Security Account Management, as the Windows OS starts up after each boot.

After each successful initial user authentication in the logging on to a non “Guest” user account, the Windows OS authentication service assigns the logged on user account to the “Authenticated User” Group ([SECURITY\_AUTHENTICATED\_USER\_RID](http://msdn.microsoft.com/en-us/library/aa379649.aspx)). Actually, the Windows OS security reference monitor (SRM) includes the [SECURITY\_AUTHENTICATED\_USER\_RID](http://msdn.microsoft.com/en-us/library/aa379649.aspx) group SID in the access token, which is being created for representing the logged on user account on the local Windows OS, upon the request from the Windows OS authentication service. However, for the logging on to the “Guest” user account, the [SECURITY\_AUTHENTICATED\_USER\_RID](http://msdn.microsoft.com/en-us/library/aa379649.aspx) group SID is not included in the access token for representing the “Guest” user account.

Finally, we should mention that the Windows OS (window terminal) session specific window logon user interface service (aka logonui.exe) could display an interactive user interface on the secure display area (i.e. the secure desktop) to let an interactive user restart the local machine, shutdown the local machine, or send the local machine to “sleep”, without requiring the user to be authenticated. By default, this interface to restart the local machine, shutdown the local machine, or send the local machine to “sleep” is available on the non-server Windows OS (i.e. Windows Vista) machines, but is not available on the server Windows OS (i.e. Windows Server 2008) machines. When logonui.exe does not make this interface available, a user needs to logon to a non “Guest” user account in order to attempt to restart the local machine, shutdown the local machine, or send the local machine to “sleep”.

### Remote access attempt over the network interfaces

In a distributed environment, it is likely that there would be non Windows OS machines or unmanaged Windows OS machines residing in the physical network of the environment. These machines would not be a member computer of the Windows OS forest/domain of the owner enterprise in this environment. Network interfaces of the member computers of the Windows OS forest/domain are visible to the non Windows OS machines or unmanaged Windows OS machines. Examples of network interfaces can be found in the sections of “Registered logon processes” and “Windows OS services needing user authentication when handling network requests”. From the practicability perspective, certain member computers of the Windows OS forest/domain may even need to provide public (i.e. non Windows OS-protected) information through some of these network interfaces. For example, a web page with a publicly known URL giving “visitors” information on how to contact the network’s IT helpdesk could be hosted on one of the member computers. Naturally, both the Kerberos and NTLM authentication protocols accept network requests without incorporating a password-based credential when supporting user authentication in the network interfaces. Obviously, we cannot assume that a Kerberos or NLTM client, attempting to gain access through the network interfaces, resides in a member computer of the Windows OS forest/domain. It is necessary to consider that such a Kerberos or NLTM client may be implemented by a 3rd party.

In the case of Kerberos, the protocol allows its application request ([AP\_REQ](http://msdn.microsoft.com/en-us/library/aa374762(VS.85).aspx)) PDU coming to a server from a client, where the request contains a null encrypted service ticket (KERB\_ENCRYPTED\_TICKET). When such an application request contains only a null encrypted service ticket, an access token of the [LsaTokenInformationNull](http://msdn.microsoft.com/en-us/library/aa378528(VS.85).aspx) token type with the user security ID (SID) of [SECURITY\_ANONYMOUS\_LOGON\_RID](http://msdn.microsoft.com/en-us/library/aa379649.aspx) is requested by the Windows OS authentication service. This access token of the [LsaTokenInformationNull](http://msdn.microsoft.com/en-us/library/aa378528(VS.85).aspx) token type therefore represents an anonymous remote user.

In the case of NTLM, the protocol allows its [MSV1\_0\_LM20\_LOGON](http://msdn.microsoft.com/en-us/library/aa378762(VS.85).aspx) request message (for network logon) coming to a server from a client, where the [MSV1\_0\_LM20\_LOGON](http://msdn.microsoft.com/en-us/library/aa378762(VS.85).aspx) request message contains an empty case sensitive or case insensitive challenge response and an empty user account name. When such a [MSV1\_0\_LM20\_LOGON](http://msdn.microsoft.com/en-us/library/aa378762(VS.85).aspx) request message contains only an empty challenge response and an empty user account name, an access token of the [LsaTokenInformationNull](http://msdn.microsoft.com/en-us/library/aa378528(VS.85).aspx) token type with the user security ID (SID) of [SECURITY\_ANONYMOUS\_LOGON\_RID](http://msdn.microsoft.com/en-us/library/aa379649.aspx) is requested by the Windows OS authentication service. This access token of the [LsaTokenInformationNull](http://msdn.microsoft.com/en-us/library/aa378528(VS.85).aspx) token type therefore represents an anonymous remote user.

Whenever an access token of the [LsaTokenInformationNull](http://msdn.microsoft.com/en-us/library/aa378528(VS.85).aspx) token type is requested to represent an anonymous remote user, the Windows OS authentication service is responsible to not assign the anonymous remote user to the “Authenticated User” Group ([SECURITY\_AUTHENTICATED\_USER\_RID](http://msdn.microsoft.com/en-us/library/aa379649.aspx)). Namely in the Windows OS authentication service’s internal “Au Add Standard Ids” function, the [SECURITY\_AUTHENTICATED\_USER\_RID](http://msdn.microsoft.com/en-us/library/aa379649.aspx) group SID is not requested for assigning to the access token of the [LsaTokenInformationNull](http://msdn.microsoft.com/en-us/library/aa378528(VS.85).aspx) token type. In addition, by default, the anonymous remote user is not assigned to the “Everyone” (i.e. World) group ([SECURITY\_WORLD\_RID](http://msdn.microsoft.com/en-us/library/aa379649.aspx)) in the Windows OS authentication service’s internal “Au Add Standard Ids” function.

### Summarizing the arguments for meeting the requirement

As a result, we can summarize the actions associated with an anonymous local or remote user as follows.

* The Windows OS protected resources, which grant an access to only a subject belonging to at least the “Authenticated User” Group, would deny the access to the “Guest” user account, which allows an anonymous local user to logon to itself.
* The Windows OS protected resources, which grant an access to only a subject belonging to at least the “Everyone” (i.e. World) group or the “Authenticated User” Group, would deny the access to an anonymous remote user.
* By default, an anonymous remote user does not possess the [POLICY\_LOOKUP\_NAMES](http://msdn.microsoft.com/en-us/library/ms721916(VS.85).aspx) right for making a request to the Windows OS authentication service for receiving translations between user account names and the corresponding security IDs (SIDs). Therefore, the anonymous remote user’s attempt to request the SID attributes for another user is denied.
* By default, an anonymous remote user does not possess the following rights for making a request to the Windows OS Local Security Account Management (SAM), so that the anonymous remote user’s attempt to enumerate the names of user accounts or group memberships, or to read the enforced password policy is denied:
  + [DOMAIN\_LIST\_ACCOUNTS](http://msdn.microsoft.com/en-us/library/cc245522.aspx);
  + [DOMAIN\_READ\_PASSWORD\_PARAMETERS](http://msdn.microsoft.com/en-us/library/cc245522.aspx);
  + [USER\_LIST\_GROUPS](http://msdn.microsoft.com/en-us/library/cc245522.aspx);
  + [GROUP\_LIST\_MEMBERS](http://msdn.microsoft.com/en-us/library/cc245522.aspx);
  + [ALIAS\_LIST\_MEMBERS](http://msdn.microsoft.com/en-us/library/cc245522.aspx).
* By default, an anonymous remote user possesses the rights for making a request to the Windows OS SMB server service for server drivers (srvsvc.dll) or a Windows OS server driver for the following purposes:
  + [Retrieving the server name and platform information in a SERVER\_INFO\_100 structure or the server name, type, and associated OS version in a SERVER\_INFO\_101 structure](http://msdn.microsoft.com/en-us/library/aa370624.aspx);
  + [Retrieving the information about a share including its share name and its share type of either disk drive (i.e. c:), print queue (i.e. LPT1:), communication device (i.e. comm), inter-process communication (i.e. IPC), and a comment associated with the share in a SHARE\_INFO\_0, SHARE\_INFO\_1, or SHARE\_INFO\_501 Structure](http://msdn.microsoft.com/en-us/library/bb525388(VS.85).aspx);
  + [Establishing a SMB connection to a non-admin share over a SMB transport protocol](http://msdn.microsoft.com/en-us/library/cc246330.aspx).
* By default, an anonymous remote user has access to the following named pipes in order to use the non security relevant RPC interfaces implemented over them:
  + netlogon, where the [netlogon RPC interfaces](http://msdn.microsoft.com/en-us/library/cc208060.aspx) are handled by the Windows OS netlogon service (netlogon.dll) for maintaining a secure channel between the local Windows OS computer and a Windows OS domain controller for authenticating users and services;
  + lsarpc, where the [LSA name translation method RPC interfaces](http://msdn.microsoft.com/en-us/library/cc234420.aspx) are handled by the Windows OS Authentication Service (aka lsass.exe);
  + samr, where the [SAM RPC interfaces](http://msdn.microsoft.com/en-us/library/cc245476.aspx) are handled by the Windows OS Local Security Account Management (SAM) (samsrv.dll);
  + browser, where the [browser auxiliary RPC interfaces](http://msdn.microsoft.com/en-us/library/cc224522.aspx) are handled by the Windows OS Common Internet File System (CIFS) Browser Auxiliary Protocol service (browser.dll).
* By default, an anonymous remote user has no access to any shares maintained by the Windows OS SMB server service for server drivers (srvsvc.dll).
* By default, on a non-server Windows OS (i.e. Windows Vista) machine, the Windows OS (window terminal) session specific window logon user interface service (aka logonui.exe) displays an interactive user interface to restart the local machine, shutdown the local machine, or send the local machine to “sleep”. However, an administrator can configure the non-server Windows OS so that the interface is hidden from the interactive users, prior to their successful logging on to a user account.
* By default, on a server Windows OS (i.e. Windows Server 2008) machine, the Windows OS (window terminal) session specific window logon user interface service (aka logonui.exe) does not display an interactive user interface to restart the local machine, shutdown the local machine, or send the local machine to “sleep”.

In conclusion, except for certain cases needing to support interoperability, or general usability, the Windows OS meets the intents of this Commercial Grade OS Requirement Set “3.1.1.1” requirement. Specific exception cases have been explained in the above.

## Addressing 3.1.1.2 “The OS shall be able to support passwords up to 32 characters in length, consisting of any combination of upper and lower case letters, numbers, and punctuations”

This requirement is addressed by the Windows OS as follows.

As mentioned earlier, the Windows OS window logon user interface service (logonui.exe) is responsible to collect a credential, through its dialog in the secure display area, from the interactive user. The dialog is implemented in the Windows Authentication UI component (authui.dll). The authui.dll component has a built-in length limit for setting to the editable UI control edit element for collecting a user account password. This length limit is 127 (LOGONUI\_PWLEN). The string read from the password edit is encrypted before passing to the Windows OS window logon state maintaining service (winlogon.exe) in a [KERB\_INTERACTIVE\_UNLOCK\_LOGON](http://msdn.microsoft.com/en-us/library/aa378088(VS.85).aspx) structure. The Windows OS window logon state maintaining service then forwards the [KERB\_INTERACTIVE\_UNLOCK\_LOGON](http://msdn.microsoft.com/en-us/library/aa378088(VS.85).aspx) structure to the Windows OS Authentication Service to attempt to complete the user account logging on.

Due to the above 127 length limit of the password UI control edit element, the Windows OS supports passwords up to 127 characters in length for interactive user account logging on.

To find evidence of the kinds of password characters supported by the Windows OS, the enforcement of the [password complexity policy (DOMAIN\_PASSWORD\_COMPLEX)](http://support.microsoft.com/kb/279890) can be examined. The policy is enforced by the Windows OS Local Security Account Management (SAM) (samsrv.dll) in its internal “Check Strong Password Restrictions” function. Among other things, the policy checks a password having a character of at least 3 of the following 5 types:

* Decimal digit characters ([C1\_DIGIT](http://msdn.microsoft.com/en-us/library/cc194821.aspx));
* Uppercase characters ([C1\_UPPER](http://msdn.microsoft.com/en-us/library/cc194821.aspx));
* Lowercase characters ([C1\_LOWER](http://msdn.microsoft.com/en-us/library/cc194821.aspx));
* Linguistic characters ([C1\_ALPHA](http://msdn.microsoft.com/en-us/library/cc194821.aspx));
* Special characters:
  + a character in the following string“`~!@#$%^&\*\_-+=|\\{}[]:;\"'<>,.”.

This shows that the Windows OS supports passwords consisting of a combination of characters from the required sets, assuming that the passwords are allowed by the password complexity policy.

Therefore, this Commercial Grade OS Requirement Set “3.1.1.2” requirement is met.

## Addressing 3.1.1.3 “The OS authentication mechanism shall provide a specific set of capabilities”

The Commercial Grade OS Requirement Set requires the following specific set of capabilities:

1. for all administrator accounts, a delay such that there can be no more than ten attempts per minute;
2. for all other accounts, a delay such that there can be no more than twenty attempts per minute.

This requirement is addressed by the Windows OS as follows.

Recall earlier, the Windows OS (window terminal) session specific window logon state maintaining service (aka winlogon.exe) is necessary in the critical path of the processing of every logon attempt of a local or remote interactive user.

There are two cases to be considered:

* the interactive user is local
  + the user attempts to logon to a user account using the locally attached input devices;
* the interactive user is remote
  + the user attempts to logon to a user account over the RDP protocol on top of the TCPIP transport through the Windows OS remote window terminal service (termsrv.dll).

The Windows OS window logon state maintaining service (winlogon.exe) maintains the following three hard-coded constants:

* LOCKOUT\_BAD\_LOGON\_COUNT (5);
* LOCKOUT\_BAD\_LOGON\_PERIOD (60 seconds);
* LOCKOUT\_BAD\_LOGON\_DELAY (30 seconds).

In the case where the interactive user is local, after the count of the consecutive failed logon to the same user account exceeds the LOCKOUT\_BAD\_LOGON\_COUNT (5) within the LOCKOUT\_BAD\_LOGON\_PERIOD (60 seconds), the Windows OS window logon state maintaining service introduces the LOCKOUT\_BAD\_LOGON\_DELAY (30 seconds) in its processing by sending the Windows OS to sleep for the LOCKOUT\_BAD\_LOGON\_DELAY (30 seconds). As a result, the eleventh attempt to logon to the same user account does not occur within the first minute after the first logon attempt. Therefore, the two capabilities specified in this Commercial Grade OS Requirement Set “3.1.1.3” requirement are met.

In the case where the interactive user is remote, after the count of the consecutive failed logon to the same user account exceeds the LOCKOUT\_BAD\_LOGON\_COUNT (5) within the LOCKOUT\_BAD\_LOGON\_PERIOD (60 seconds), the Windows OS window logon state maintaining service induces the Windows OS remote window terminal service (termsrv.dll) to disconnect the RDP connection by shutting itself down. As a result, the eleventh attempt to logon to the same user account does not occur within the first minute after the first logon attempt because the establishment of a new RDP connection and a starting up of the Windows OS window logon state maintaining service take longer than a minute. Therefore, the two capabilities specified in this Commercial Grade OS Requirement Set “3.1.1.3” requirement are met.

## Addressing 3.1.1.4 “The OS shall provide only obscured feedback to the user while the authentication is in progress”

This requirement is addressed by the Windows OS as follows.

As mentioned earlier, the Windows OS window logon user interface service (logonui.exe) is responsible to collect a credential, through its dialog in the secure display area, from the interactive user. The dialog is implemented in the Windows Authentication UI component (authui.dll). The authui.dll component uses a special editable UI control edit element for collecting a user account password or smart PIN. This special control edit element treats user input in the following specific way.

After the user types in a character from his/her keyboard or selects a character from his/her input device, the special control edit element requests the Windows OS window manager (namely win32k.sys) to redraw all visible characters in the control edit element using a special character (such as the asterisk (\*)) instead of the actual character that the user has input. The edit element repeats this step for every character that the user inputs for his/her credential. As a result, the user feedback for his/her input is obscured when appearing in the edit element.

Therefore, this Commercial Grade OS Requirement Set “3.1.1.4” requirement is met.

## Addressing 3.1.1.5 “The OS shall detect when an authorized administrator specified positive integer of consecutive unsuccessful authentication attempts occur related to any authorized user authentication process”

This requirement is addressed by the Windows OS as follows.

The Windows OS Local Security Account Management (SAM) (samsrv.dll) is responsible (through its internal “Increment Bad Password Count” function) to update the bad password count of a specific user account. The [LockoutObservationWindow](http://msdn.microsoft.com/en-us/library/cc245830.aspx) is the administrator configured time period in which bad password attempts are counted without resetting the count to zero. If the current time is still inside the [LockoutObservationWindow](http://msdn.microsoft.com/en-us/library/cc245830.aspx) since the start of the [LockoutObservationWindow](http://msdn.microsoft.com/en-us/library/cc245830.aspx) due to an occurrence of a bad password attempt for a specific user account, the SAM increments the bad password count of the user account. If the incremented bad password count reaches the administrator configured [LockoutThreshold](http://msdn.microsoft.com/en-us/library/cc245830.aspx), then SAM conducts the following:

* Locking out the specific user account by setting the [USER\_ACCOUNT\_AUTO\_LOCKED](http://msdn.microsoft.com/en-us/library/cc211789.aspx) code value to the [user account control attribute](http://msdn.microsoft.com/en-us/library/ms680832(VS.85).aspx) of the user account;
* Urgently replicating the last lockout time of the specific user account as the current time to other Windows OS domain controllers in the case where the user account is a domain-wide user account;
* Generating the [Event ID 4740](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ACCOUNT\_AUTO\_LOCKED/SE\_AUDITID\_ETW\_ACCOUNT\_AUTO\_LOCKED) “A user account was locked out” audit record to record the event that the specific user account is locked out, in the case where the user account was not previously locked.

If the current time is outside the [LockoutObservationWindow](http://msdn.microsoft.com/en-us/library/cc245830.aspx) since the start of the [LockoutObservationWindow](http://msdn.microsoft.com/en-us/library/cc245830.aspx) due to an occurrence of a bad password attempt for a specific user account, the SAM resets the bad password count of the user account to zero and restarts a next [LockoutObservationWindow](http://msdn.microsoft.com/en-us/library/cc245830.aspx). Additionally, the SAM also resets the bad password count of the user account to zero, when a successful logon using the password is reported.

The update of the bad password count of a specific user account occurs when

* a password-capable package (such as Kerberos, NTLM or Digest), mentioned in the “Authentication Protocols” section reports to the SAM with one of the following indicators:
  + USER\_LOGON\_BAD\_PASSWORD, indicating a bad password attempt;
  + USER\_LOGON\_NET\_SUCCESS\_LOGON, indicating a successful network logon;
  + USER\_LOGON\_INTER\_SUCCESS\_LOGON, indicating a successful interactive logon;
* the SAM handles a request to change the password of a specific user account, where the supplied old password extracted from the request is bad;
* the SAM handles a request to change the password of a specific user account, where the supplied old password extracted from the request is good.

Therefore, the above shows that the Windows OS detects when an authorized administrator specified positive integer of consecutive unsuccessful authentication attempts occur related to an authorized user authentication process. Specifically, the [Event ID 4740](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ACCOUNT\_AUTO\_LOCKED/SE\_AUDITID\_ETW\_ACCOUNT\_AUTO\_LOCKED) “A user account was locked out” audit record is generated.

Therefore, this Commercial Grade OS Requirement Set “3.1.1.5” requirement is met.

## Addressing 3.1.1.6 “The OS shall perform a specific set of actions when the defined number of consecutive unsuccessful authentication attempts specified in “3.1.1.5” has been detected”

The Commercial Grade OS Requirement Set requires the following specific set of actions being performed by the OS:

1. for all administrator accounts, disable the account for an authorized administrator configurable time period;
2. for all other accounts, disable the user account until it is re-enabled by an authorized administrator;
3. for all disable accounts, respond with an “account disabled” message without attempting any type of authentication.

This requirement is addressed by the Windows OS as follows.

Every password-capable security provider (such as Kerberos, NTLM or Digest), mentioned in the “Authentication Protocols” section checks the existence of the USER\_ACCOUNT\_AUTO\_LOCKED code value in the [user account control attribute](http://msdn.microsoft.com/en-us/library/ms680832(VS.85).aspx) of the user account being attempted for logging on. If the USER\_ACCOUNT\_AUTO\_LOCKED code value exists, the following behaviors occur regardless of the validity of the supplied password.

* The NTLM security provider fails the logon attempt by returning [STATUS\_ACCOUNT\_LOCKED\_OUT](http://msdn.microsoft.com/en-us/library/cc704588.aspx) in the following situations:
  + the user account being attempted for logging on is not the built-in administrator ([DOMAIN\_USER\_RID\_ADMIN](http://msdn.microsoft.com/en-us/library/aa379649.aspx));
  + the user account being attempted for logging on is the built-in administrator ([DOMAIN\_USER\_RID\_ADMIN](http://msdn.microsoft.com/en-us/library/aa379649.aspx)), the logging on attempt is not interactive, and the administrator configured domain password policy of the [DOMAIN\_LOCKOUT\_ADMINS](http://msdn.microsoft.com/en-us/library/aa375371(VS.85).aspx)[[6]](#footnote-7) is set.
* As the Digest security provider does not handle the attempt of a logging on to the built-in administrator account, the Digest security provider fails the logon attempt by returning [STATUS\_ACCOUNT\_LOCKED\_OUT](http://msdn.microsoft.com/en-us/library/cc704588.aspx).
* As the Kerberos security provider does not handle the attempt of a logging on to the built-in administrator account, the Kerberos security provider fails the logon attempt by returning [STATUS\_ACCOUNT\_LOCKED\_OUT](http://msdn.microsoft.com/en-us/library/cc704588.aspx).

As a result, Action a) of this requirement is met except the interactive logging on attempt to the built-in administrator account. For the built-in administrator account logging on attempt, there are still the following behaviors (as explained in rationale to address the Commercial Grade OS Requirement Set “3.1.1.3” requirement) to counter the same threat that Action a) is intended to address.

* In the case where the interactive user is local, after the count of the consecutive failed logon to the built-in administrator account exceeds the LOCKOUT\_BAD\_LOGON\_COUNT (5) within the LOCKOUT\_BAD\_LOGON\_PERIOD (60 seconds), the Windows OS window logon state maintaining service introduces the LOCKOUT\_BAD\_LOGON\_DELAY (30 seconds) in its processing by sending the Windows OS to sleep for the LOCKOUT\_BAD\_LOGON\_DELAY (30 seconds).
* In the case where the interactive user is remote, after the count of the consecutive failed logon to the built-in administrator account exceeds the LOCKOUT\_BAD\_LOGON\_COUNT (5) within the LOCKOUT\_BAD\_LOGON\_PERIOD (60 seconds), the Windows OS window logon state maintaining service induces the Windows OS remote window terminal service (termsrv.dll) to disconnect the RDP connection by shutting itself down.

In the case, where the failed logging on attempt is interactive and the user account being attempted for logging on is not the built-in administrator, the window logon user interface service displays the following message in the secure display area, due to the error code of [STATUS\_ACCOUNT\_LOCKED\_OUT](http://msdn.microsoft.com/en-us/library/cc704588.aspx).

“The user account has been automatically locked because too many invalid logon attempts or password change attempts have been requested.”

As a result, Action c) of this requirement is met.

In the case, where the failed logging on attempt is interactive and the user account being attempted for logging on is the built-in administrator, the interactive user may attempt another logging on to the built-in administrator account with another password, without receiving the above auto locked out dialog message.

Note that the [DOMAIN\_LOCKOUT\_ADMINS](http://msdn.microsoft.com/en-us/library/aa375371(VS.85).aspx) domain password policy is not enabled by default. If an administrator wishes to enable it, s/he could specify this policy setting using the DOMAIN\_PASSWORD\_INFORMATION structure through the [SamrSetInformationDomain()](http://msdn.microsoft.com/en-us/library/cc245788.aspx) interface of the SAM. Unfortunately, there is not a supported tool available for an administrator to set this [DOMAIN\_LOCKOUT\_ADMINS](http://msdn.microsoft.com/en-us/library/aa375371(VS.85).aspx) domain password policy, without writing at least a programming language specific script.

The [LockoutDuration](http://msdn.microsoft.com/en-us/library/cc232773.aspx) domain policy value is administrator-configurable. It indicates the duration for which a user account remains locked out (i.e. the USER\_ACCOUNT\_AUTO\_LOCKED code value remaining set) before being automatically reset to an unlocked state (i.e. the USER\_ACCOUNT\_AUTO\_LOCKED code value being cleared). This [LockoutDuration](http://msdn.microsoft.com/en-us/library/cc232773.aspx) value is checked by the SAM when the SAM updates the bad password count of a specific user account. If the current time has passed the [LockoutDuration](http://msdn.microsoft.com/en-us/library/cc232773.aspx) value since the last lockout time, then the SAM clears the USER\_ACCOUNT\_AUTO\_LOCKED code value in the [user account control attribute](http://msdn.microsoft.com/en-us/library/ms680832(VS.85).aspx) of the user account.

An administrator is provided the permission to clear the USER\_ACCOUNT\_AUTO\_LOCKED code value in the [user account control attribute](http://msdn.microsoft.com/en-us/library/ms680832(VS.85).aspx) of a given user account. The administrator can accomplish this task for the user account through either the SAM interface or through the Windows OS LDAP server. After the task is accomplished, the [Event ID 4767](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ACCOUNT\_UNLOCKED/SE\_AUDITID\_ETW\_ACCOUNT\_UNLOCKED) “A user account was unlocked” audit record to record the event that the specific user account is unlocked is generated.

If the [LockoutDuration](http://msdn.microsoft.com/en-us/library/cc232773.aspx) domain policy value is zero, then a locked out user account remains locked out indefinitely until an administrator explicitly unlocks the user account [by clearing the account’s USER\_ACCOUNT\_AUTO\_LOCKED code value](http://msdn.microsoft.com/en-us/library/ms696026(VS.85).aspx).

As a result, Action b) of this requirement is met.

In conclusion, except for the certain case needing to support general usability and to reduce the exposure of a denial-of-service by allowing the built-in administrator to re-attempt a logon interactively, the Windows OS meets the intents of this requirement. The above has explained the exception case in detail.

An alternative way to meet this requirement is to disable the built-in administrator account all together (i.e. setting USER\_ACCOUNT\_DISABLED in the [user account control attribute](http://msdn.microsoft.com/en-us/library/ms680832(VS.85).aspx) of the built-in administrator account), using the security option of “[Accounts: Administrator account status](http://technet2.microsoft.com/windowsserver/en/library/4d4322d8-6e32-411f-9091-3941d6a90bb91033.mspx?mfr=true)”. By default, the built-in administrator account is disabled on Windows Vista and is enabled on Windows Server 2008. When the built-in administrator account is disabled, the inability to lock out the built-in administrator account is immaterial.

Therefore, this Commercial Grade OS Requirement Set “3.1.1.6” requirement is addressed.

## Addressing 3.1.1.7 “The OS shall maintain a specific list of security attributes belonging to individual users”

The Commercial Grade OS Requirement Set requires the following specific list of security attributes belonging to individual users being maintained by the OS:

1. unique user identity;
2. group memberships;
3. authentication data;
4. any other security-relevant authorizations or attributes (e.g. roles).

This requirement is addressed by the Windows OS as follows.

The Windows OS defines user account objects in the Active Directory (for Windows OS domain wide user accounts) and in the SAM of an individual machine (for Windows OS local user accounts on the machine). In either case, the following lists a subset of attributes that are defined for user account objects:

* [ATT\_NT\_SECURITY\_DESCRIPTOR](http://msdn.microsoft.com/en-us/library/ms679006(VS.85).aspx) (if the user account is domain based) or the security descriptor for the user account (if the user account is defined on a local machine)
  + It defines the access control security descriptor for gaining access to the user account attributes;
  + It also specifies the auditing requirement when an access to the attributes occurs;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_SAM\_ACCOUNT\_NAME](http://msdn.microsoft.com/en-us/library/ms679635(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_DISPLAY\_NAME](http://msdn.microsoft.com/en-us/library/ms675514(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_DESCRIPTION](http://msdn.microsoft.com/en-us/library/ms675492(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_USER\_COMMENT](http://msdn.microsoft.com/en-us/library/ms676199(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_USER\_PARAMETERS](http://msdn.microsoft.com/en-us/library/ms680847(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_HOME\_DIRECTORY](http://msdn.microsoft.com/en-us/library/ms676190(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It specifies the UNC path to the home directory for the user account;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_HOME\_DRIVE](http://msdn.microsoft.com/en-us/library/ms676191(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_SCRIPT\_PATH](http://msdn.microsoft.com/en-us/library/ms679656(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It stores the path to the [logon script](http://technet2.microsoft.com/windowsserver/en/library/12cedcb6-a076-461d-bc73-0cc4048eff0d1033.mspx?mfr=true) for the user account so that certain policy elements of the user account are enforced, through the execution of the script, on the logon machine after the logging on to the user account;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_PROFILE\_PATH](http://msdn.microsoft.com/en-us/library/ms679422(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It stores the path to the [user profile](http://technet2.microsoft.com/windowsserver/en/library/093238f3-5064-470e-a281-0eb1c28b9cf01033.mspx?mfr=true) for the user account so that certain policy elements of the user account are enforced, through the configurations made in the profile, on the logon machine after the logging on to the user account;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It specifies the [hours](http://technet2.microsoft.com/windowsserver/en/library/a0f007ba-b3ef-4c1f-8836-9087fe5eb08d1033.mspx?mfr=true) that the user account is allowed to be logged on in the Windows OS domain;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_IS\_MEMBER\_OF\_DL](http://msdn.microsoft.com/en-us/library/ms677099(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It contains the distinguished names of the groups to which this user account directly belongs to;
  + Group nesting is possible in Windows OS;
  + A [group object](http://msdn.microsoft.com/en-us/library/ms682251(VS.85).aspx) also has its own ATT\_IS\_MEMBER\_OF\_DL attribute and so SAM is responsible to conduct the group membership expansion;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_DBCS\_PWD](http://msdn.microsoft.com/en-us/library/ms675480(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + this stores the LM OWF password and it is not relevant because the security option of “[Network security: Do not store LAN Manager hash value on next password change](http://technet2.microsoft.com/windowsserver/en/library/393fa32d-04dd-4a15-b23d-3fc2b85588821033.mspx?mfr=true)” is enabled by default;
* [ATT\_UNICODE\_PWD](http://msdn.microsoft.com/en-us/library/ms680513(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It is not readable from outside the Windows OS process for SAM and it is not writable by a subject directly;
  + It stores the NTLM password hash in the format suitable for the user authentication being conducted by the NTLM security provider;
  + If the user account is domain based, it can be updated by a subject with the proven knowledge of the current password, in the case where there are the ACCESS\_ALLOWED\_OBJECT\_ACE ACEs and not the ACCESS\_DENIED\_OBJECT\_ACE ACEs for the GUID\_CONTROL\_UserChangePassword property set to allow and not deny everyone and the principal self in the user account object’s security descriptor;
  + If the user account is locally defined, it can be updated by a subject with the proven knowledge of the current password, in the case where there are ACCESS\_ALLOWED\_ACE ACEs for USER\_CHANGE\_PASSWORD to allows everyone and the principal self in the user account object’s security descriptor;
  + If the user account is domain based, it can be updated by an administrator without the knowledge of the current password as there is the ACCESS\_ALLOWED\_OBJECT\_ACE ACE for the GUID\_CONTROL\_UserForceChangePassword property set to allow the administrator in the user account object’s default security descriptor;
  + If the user account is locally defined, it can be updated by an administrator without the knowledge of the current password as there is the ACCESS\_ALLOWED\_ACE ACE for USER\_FORCE\_PASSWORD\_CHANGE to allow the administrator in the user account object’s default security descriptor;
  + It is encrypted when residing in the database of SAM or of Active Directory [according to the system key (syskey) policy](http://support.microsoft.com/kb/818200);
  + Basically, the syskey policy specifies that the master key used for the encryption is either
    - stored locally under the “HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa” registry key as certain key values after running the master key through a specific obfuscation algorithm;
    - supplied by an interactive user on the secure display area during machine boot as the MD5 hash of a password being entered by the user through the interactive dialog, captioned “Startup Password”;
    - supplied by an interactive user on the secure display area during machine boot as the string read from a removable disk being inserted by the user due to the interactive dialog, captioned “Startup Key Disk”;
    - the master key is available to only the following components:
      * the window boot (aka wininit.exe) or the first instance of the window logon state maintaining service (aka winlogon.exe);
      * the Windows OS Authentication Service (aka lsass.exe);
      * the Windows OS Local Security Account Management (SAM), which also resides inside lsass.exe;
* [ATT\_NT\_PWD\_HISTORY](http://msdn.microsoft.com/en-us/library/ms679004(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It is not readable outside the Windows OS process for SAM and it is not writable by a subject directly;
  + It is encrypted when residing in the database of SAM or of Active Directory [according to the system key (syskey) policy](http://support.microsoft.com/kb/818200);
* [ATT\_LM\_PWD\_HISTORY](http://msdn.microsoft.com/en-us/library/ms676833(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + this stores the LM OWF password history and it is not relevant because the security option of “[Network security: Do not store LAN Manager hash value on next password change](http://technet2.microsoft.com/windowsserver/en/library/393fa32d-04dd-4a15-b23d-3fc2b85588821033.mspx?mfr=true)” is enabled by default;
* [ATT\_LAST\_LOGON](http://msdn.microsoft.com/en-us/library/ms676823(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_PWD\_LAST\_SET](http://msdn.microsoft.com/en-us/library/ms679430(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_ACCOUNT\_EXPIRES](http://msdn.microsoft.com/en-us/library/ms675098(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_BAD\_PASSWORD\_TIME](http://msdn.microsoft.com/en-us/library/ms675243(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_RID](http://msdn.microsoft.com/en-us/library/ms679456(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_USER\_ACCOUNT\_CONTROL](http://msdn.microsoft.com/en-us/library/ms680832(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + Among other code values, it contains specifically the USER\_ACCOUNT\_AUTO\_LOCKED, USER\_ACCOUNT\_DISABLED code values for supporting the account lock out policy described earlier;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_COUNTRY\_CODE](http://msdn.microsoft.com/en-us/library/ms675466(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_CODE\_PAGE](http://msdn.microsoft.com/en-us/library/ms675451(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_BAD\_PWD\_COUNT](http://msdn.microsoft.com/en-us/library/ms675244(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It stores the number of times a user tried to consecutively log on to the user account using an incorrect password;
  + It is set by SAM and is not writable from outside the Windows OS process for SAM;
* [ATT\_LOGON\_COUNT](http://msdn.microsoft.com/en-us/library/ms676845(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_SAM\_ACCOUNT\_TYPE](http://msdn.microsoft.com/en-us/library/ms679637(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_LOCAL\_POLICY\_FLAGS](http://msdn.microsoft.com/en-us/library/ms676837(vs.85).aspx) (if the user account is domain based);
* [ATT\_SUPPLEMENTAL\_CREDENTIALS](http://msdn.microsoft.com/en-us/library/ms679920(VS.85).aspx) (if the user account is domain based)
  + It is not readable outside the Windows OS process for SAM and it is not writable by a subject directly;
  + It [stores](http://msdn.microsoft.com/en-us/library/cc211940.aspx) the password-derived keys, the password hashes, or the clear text password in a format suitable for the user authentication being conducted by the Kerberos KDC and Kerberos security provider and the Digest security provider;
  + The Clear text password is stored only if the “[Store passwords using reversible encryption](http://technet2.microsoft.com/WindowsServer/en/Library/eeff044c-d4a8-4699-a4b8-c5e563118c931033.mspx?mfr=true)” domain password policy is enabled or the user account is configured to “[support per user reversible encrypted password storage (i.e. USER\_ENCRYPTED\_TEXT\_PASSWORD\_ALLOWED)](http://msdn.microsoft.com/en-us/library/cc245673.aspx)”;
  + By default, the “[Store passwords using reversible encryption](http://technet2.microsoft.com/WindowsServer/en/Library/eeff044c-d4a8-4699-a4b8-c5e563118c931033.mspx?mfr=true)” domain password policy is disabled;
  + By default, a user account is not configured to “[support per user reversible encrypted password storage (i.e. USER\_ENCRYPTED\_TEXT\_PASSWORD\_ALLOWED)](http://msdn.microsoft.com/en-us/library/cc245673.aspx)”;
  + It is encrypted when residing in the database of SAM or of Active Directory [according to the system key (syskey) policy](http://support.microsoft.com/kb/818200);
* [ATT\_SID\_HISTORY](http://msdn.microsoft.com/en-us/library/ms679833(VS.85).aspx) (if the user account is domain based)
  + It contains previous SIDs used for the user account if the user account was moved from another Windows OS domain;
  + Whenever an object is moved from one domain to another, a new SID is created and that new SID becomes the value of the object’s ATT\_OBJECT\_SID attribute;
  + It is set by SAM and is not writable from outside the Windows OS process for SAM;
* [ATT\_LOCKOUT\_TIME](http://msdn.microsoft.com/en-us/library/ms676843(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + Its value supports the account lock out policy mentioned earlier;
  + It is set by SAM and is not writable from outside the Windows OS process for SAM;
* [ATT\_IS\_CRITICAL\_SYSTEM\_OBJECT](http://msdn.microsoft.com/en-us/library/ms676798(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_USER\_PRINCIPAL\_NAME](http://msdn.microsoft.com/en-us/library/ms680857(VS.85).aspx) (if the user account is domain based);
* [ATT\_MS\_DS\_CREATOR\_SID](http://msdn.microsoft.com/en-us/library/ms678637(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_OBJECT\_SID](http://msdn.microsoft.com/en-us/library/ms679024(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It stores the security identifier (SID) of the user account;
  + It is set by the SAM and is not writable from outside the Windows OS process for the SAM;
  + The structure of a SID is [a variable-length byte array that uniquely identifies a security principal (i.e. a user or computer account, or a group)](http://msdn.microsoft.com/en-us/library/cc230371.aspx);
  + The SID is issued and assigned when the security principal is created;
  + The first integer in the SID array contains a value that is unique relative to the [IdentifierAuthority](http://msdn.microsoft.com/en-us/library/cc230372(PROT.10).aspx) of the SID, and all subsequent integers are unique relative to the previous integer;
  + The [IdentifierAuthority](http://msdn.microsoft.com/en-us/library/cc230372(PROT.10).aspx) of the SID, or SubAuthority[n-1], “issues” SubAuthority[n], for n>0;
  + Each integer is referred to as a RID (relative identifier), and the last RID in the array uniquely identifies the security principal in the domain specified by SubAuthorities, 0 through n-1;
  + The responsible RID “issuer” in a Windows OS domain is the [RID flexible single-master operations (FSMO) owner](http://msdn.microsoft.com/en-us/library/cc200970.aspx), which issues a monotonically-increasing RID from its allocated pool of RIDs;
  + Due to the RID monotonic-increasing characteristic and the SID array structure, unique SIDs (as user identities) for domain based user accounts are maintained in the whole life cycle of the Windows OS, as required;
  + In the case of a locally defined user account, there is no [RID flexible single-master operations (FSMO) owner](http://msdn.microsoft.com/en-us/library/cc200970.aspx), and the local Windows OS SAM simply picks the next free and higher RID from the list of sequential RIDs that it maintains in its database residing within the Windows OS registry;
  + Due to the increasing RID sequence maintained in the local Windows OS SAM’s database and the SID array structure, unique SIDs (as user identities) for locally defined user accounts are maintained in the whole life cycle of the Windows OS, as required;
* [ATT\_MS\_DS\_SITE\_AFFINITY](http://msdn.microsoft.com/en-us/library/ms677824(VS.85).aspx) (if the user account is domain based);
* [ATT\_LAST\_LOGON\_TIMESTAMP](http://msdn.microsoft.com/en-us/library/ms676824(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account;
* [ATT\_MS\_DS\_CACHED\_MEMBERSHIP](http://msdn.microsoft.com/en-us/library/ms677402(VS.85).aspx) (if the user account is domain based)
  + It stores the cached group memberships of the user account while still within the [cached membership staleness time](http://technet2.microsoft.com/WindowsServer/en/Library/0d34c3b9-499b-41d3-a55f-527ce61e78581033.mspx?mfr=true);
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_MS\_DS\_CACHED\_MEMBERSHIP\_TIME\_STAMP](http://msdn.microsoft.com/en-us/library/ms677403(VS.85).aspx) (if the user account is domain based);
* [ATT\_MS\_DS\_USER\_ACCOUNT\_CONTROL\_COMPUTED](http://msdn.microsoft.com/en-us/library/ms677840(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It is set by the SAM and is not writable from outside the Windows OS process for the SAM;
* [ATT\_USER\_PASSWORD](http://msdn.microsoft.com/en-us/library/ms680851(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It is not [used](http://msdn.microsoft.com/en-us/library/cc200470.aspx) by default, unless the Windows OS server is deployed in the [Active Directory Application Mode (ADAM)](http://technet2.microsoft.com/WindowsServer/en/library/29fb059e-544c-4577-bf7c-ba4b08df48431033.mspx?mfr=true) for 3rd party applications which need a clear-text password for their users;
* [ATT\_MS\_DS\_ALLOWED\_TO\_DELEGATE\_TO](http://msdn.microsoft.com/en-us/library/ms677183(VS.85).aspx) (if the user account is domain based)
  + It is relevant only if the user account represents an identity to run a service;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_SERVICE\_PRINCIPAL\_NAME](http://msdn.microsoft.com/en-us/library/ms679785(VS.85).aspx) (if the user account is domain based)
  + It is relevant only if the user account represents an identity to run a service;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_MS\_DS\_KEYVERSIONNUMBER](http://msdn.microsoft.com/en-us/library/ms677432(VS.85).aspx) (if the user account is domain based);
* [ATT\_DNS\_HOST\_NAME](http://msdn.microsoft.com/en-us/library/ms675524(VS.85).aspx) (if the user account is domain based);
* [ATT\_MS\_DS\_SECONDARY\_KRBTGT\_NUMBER](http://msdn.microsoft.com/en-us/library/ms677816(VS.85).aspx) (if the user account is domain based);
* [ATT\_MS\_DS\_SUPPORTED\_ENCRYPTION\_TYPES](http://msdn.microsoft.com/en-us/library/ms677827(VS.85).aspx) (if the user account is domain based);
  + It specifies the encryption algorithms supported by the user account;
  + The Kerberos KDC uses the information in this attribute while generating a service ticket for the user account;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) (if the user account is domain based)
  + It contains associated mappings for X.509 public key certificates or external Kerberos user accounts to the user account for the purpose of public key certificate or smart card based user authentication;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_MS\_DS\_AUTHENTICATEDAT\_DC](http://msdn.microsoft.com/en-us/library/ms677186(VS.85).aspx) (if the user account is domain based);
* [ATT\_MS\_DS\_LAST\_SUCCESSFUL\_INTERACTIVE\_LOGON\_TIME](http://msdn.microsoft.com/en-us/library/ms677436(VS.85).aspx) (if the user account is domain based)
  + It is set by SAM and is not writable from outside the Windows OS process for SAM;
  + It supports the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy when the policy is enabled;
* [ATT\_MS\_DS\_LAST\_FAILED\_INTERACTIVE\_LOGON\_TIME](http://msdn.microsoft.com/en-us/library/ms677435(VS.85).aspx) (if the user account is domain based);
  + It is set by SAM and is not writable from outside the Windows OS process for SAM;
  + It supports the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy when the policy is enabled;
* [ATT\_MS\_DS\_FAILED\_INTERACTIVE\_LOGON\_COUNT](http://msdn.microsoft.com/en-us/library/ms677418(VS.85).aspx) (if the user account is domain based);
  + It is set by SAM and is not writable from outside the Windows OS process for SAM;
  + It supports the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy when the policy is enabled;
* [ATT\_MS\_DS\_FAILED\_INTERACTIVE\_LOGON\_COUNT\_AT\_LAST\_SUCCESSFUL\_LOGON](http://msdn.microsoft.com/en-us/library/ms677419(VS.85).aspx) (if the user account is domain based)
  + It is set by SAM and is not writable from outside the Windows OS process for SAM;
  + It supports the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy when the policy is enabled.

The above Windows OS attributes of a user account object encapsulate the necessary security attributes belonging to individual users. As these Windows OS attributes are maintained in the Active Directory (for Windows OS domain wide user accounts) and in the SAM of an individual machine (for Windows OS local user accounts on the machine), they are considered as being maintained by the OS as stated in this Commercial Grade OS Requirement Set “3.1.1.7” requirement.

## Addressing 3.1.1.8 “The OS shall associate a specific list of user security attributes with subjects acting on behalf of that user”

The Commercial Grade OS Requirement Set requires the following specific list of user security attributes for associating with subjects acting on behalf of that user:

1. the unique user identity;
2. any group identity or identities;
3. any other security-relevant authorizations or attributes (e.g. roles).

This requirement is addressed by the Windows OS as follows.

We recall that an access token for a logged on user account is necessary to associate with any subject acting on behalf of the user who has logged on to the user account.

In the case where the Windows OS authentication service has decided a successful authentication of the user to the specified user account, with the assistance from a Windows OS credential verification specific component, the Windows OS authentication service requests the Windows OS security reference monitor (SRM) to create an access token for the logged on user account. Among other elements, the access token includes

* the user’s unique security ID (SID), which represents the logged on user account;
* the list of groups (in the form of group object SIDs) that the logged on user account is a member of directly or indirectly;
* the list of Windows OS privileges assigned to the logged on user account.

The user SID of the logged on user account is received by the responsible Windows OS credential verification specific component of the Windows OS authentication service (i.e. the corresponding security provider) from either

* the local Windows OS Local Security Account Management (SAM), in the case where the user account is a local user account defined in the local Windows OS running on the machine where the logging on occurs;
* a Windows OS domain controller (collocating with a Kerberos key distribution center and Active Directory), in the case where the user account is a domain-wide user account defined in the Windows OS-enabled distributed OS environment, which includes the local machine of the logging on user as a member computer.

Similarly, part of the group membership of the logged on user account, in the form of a list of group object SIDs after the group membership expansion, is also received by the responsible Windows OS credential verification specific component of the Windows OS authentication service (i.e. the corresponding security provider) from either

* the local Windows OS Local Security Account Management (SAM), in the case where the user account is a local user account defined in the local Windows OS running on the machine where the logging on occurs;
* a Windows OS domain controller (collocating with a Kerberos key distribution center (KDC) and Active Directory), in the case where the user account is a domain-wide user account defined in the Windows OS-enabled distributed OS environment, which includes the local machine of the logging on user as a member computer.

In the case where a Windows OS domain controller is involved, the user SID and the partial group membership of the logged on user account are packaged in a cryptographically protected structure known as the Kerberos [privilege attribute certificate (PAC)](http://msdn.microsoft.com/en-us/library/aa302203.aspx) when traveling over the network. The PAC originates from the Kerberos KDC of the Windows OS domain controller and it is consumed by the Kerberos security provider of the local machine of the logging on user. The NTLM security provider of the local machine of the logging on user also has a direct secure channel to the Windows OS domain controller for obtaining the user SID and the partial group membership. Other security providers, collocating with the Kerberos security provider and the NTLM security provider, can contact the Windows OS domain controller indirectly through the Kerberos security provider or the NTLM security provider.

After the partial group membership received from the responsible Windows OS credential verification specific component of the Windows OS authentication service (i.e. the corresponding security provider), the Windows OS authentication service finalizes the fully expanded group membership by

* considering the logon type(s) for adding the corresponding standard group memberships such as [SECURITY\_INTERACTIVE\_RID](http://msdn.microsoft.com/en-us/library/aa379649(VS.85).aspx), SECURITY\_REMOTE\_LOGON\_RID, [SECURITY\_NETWORK\_RID](http://msdn.microsoft.com/en-us/library/aa379649(VS.85).aspx), [SECURITY\_BATCH\_RID](http://msdn.microsoft.com/en-us/library/aa379649(VS.85).aspx), [SECURITY\_SERVICE\_RID](http://msdn.microsoft.com/en-us/library/aa379649(VS.85).aspx), [SECURITY\_TERMINAL\_SERVER\_RID](http://msdn.microsoft.com/en-us/library/aa379649(VS.85).aspx), or [SECURITY\_AUTHENTICATED\_USER\_RID](http://msdn.microsoft.com/en-us/library/aa379649(VS.85).aspx) as appropriate;
* adding local groups as provided by the trusted service subject who requests the Windows OS authentication service to conduct the authentication of the user who attempts the logging on to the specific user account through the interfaces of the trusted service subject:
  + for example, [SECURITY\_LOCAL\_RID](http://msdn.microsoft.com/en-us/library/aa379649(VS.85).aspx) and a unique logon ID for ensuring that only processes in a given logon session can gain access to the windowstation objects for that logon session are provided by the window logon state maintaining service to the Windows OS authentication service;
* checking with the local SAM (through [SamrGetAliasMembership()](http://msdn.microsoft.com/en-us/library/cc245816.aspx)) for any additional group membership belonging to the local groups, which are defined in the local Windows OS running on the machine where the logging on occurs, for the final group membership expansion.

Finally, the Windows OS authentication service determines the Windows OS privilege set and the system access for the logging on user account due to the user SID and the expanded final group membership of the user account from its policy database (which is maintained within its own Windows OS process).

The following is the list of Windows OS privileges that may be assigned to a user account or a group in a local Windows OS machine:

* [SeAssignPrimaryTokenPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, local service, network service;
* [SeAuditPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, local service, network service;
* [SeBackupPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group, the “Backup Operators” group on Windows Vista and Server 2008;
  + By default, it is assigned to local system, the “Administrators” group, the “Backup Operators” group, the “Server Operators” group on Windows Server 2008 domain controllers;
* [SeChangeNotifyPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group, the “Backup Operators” group, the “Users” group, the “Everyone” (i.e. World) group, local service, network service on Windows Vista and Server 2008;
  + By default, it is assigned to local system, the “Administrators” group, the “Authenticated Users” group, the “Everyone” (i.e. World) group, local service, network service, the “Pre-Windows 2000 Compatible Access” group on Windows Server 2008 domain controllers;
* [SeCreateGlobalPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, Windows OS services, the “Administrators” group, local service, network service;
* [SeCreatePagefilePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group;
* [SeCreatePermanentPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system;
* [SeCreateSymbolicLinkPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group;
* [SeCreateTokenPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system;
* [SeDebugPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group;
* [SeImpersonatePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, Windows OS services, the “Administrators” group, local service, network service;
* [SeIncreaseBasePriorityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group;
* [SeIncreaseQuotaPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group, local service, network service;
* [SeIncreaseWorkingSetPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Users” group;
* [SeLoadDriverPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group;
* [SeLockMemoryPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system;
* [SeMachineAccountPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is not assigned to any subject;
* [SeManageVolumePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group;
* [SeProfileSingleProcessPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group;
* [SeRemoteShutdownPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to the “Administrators” group on Windows Vista and Server 2008;
  + By default, it is assigned to the “Administrators” group, the “Server Operators” group on Windows Server 2008 domain controllers;
* [SeRestorePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group, the “Backup Operators” group on Windows Vista and Server 2008;
  + By default, it is assigned to local system, the “Administrators” group, the “Backup Operators” group, the “Server Operators” group on Windows Server 2008 domain controllers;
* [SeSecurityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group;
* [SeShutdownPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group, the “Backup Operators” group, the “Users” group on Windows Vista;
  + By default, it is assigned to local system, the “Administrators” group, the “Backup Operators” group on Windows Server 2008;
  + By default, it is assigned to local system, the “Administrators” group, the “Backup Operators” group, the “Server Operators” group, the “Print Operators” group on Windows Server 2008 domain controllers;
* [SeSystemEnvironmentPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group;
* [SeSystemProfilePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group;
* [SeSystemTimePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group, local service on Windows Vista and Windows Server 2008;
  + By default, it is assigned to local system, the “Administrators” group, the “Server Operators” group, local service on Windows Server 2008 domain controllers;
* [SeTakeOwnershipPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group;
* [SeTcbPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system;
* [SeTimeZonePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group, local service, the “Users” group on Windows Vista;
  + By default, it is assigned to local system, the “Administrators” group, local service on Windows Server 2008;
* [SeUndockPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system, the “Administrators” group, the “Users” group on Windows Vista;
  + By default, it is assigned to local system, the “Administrators” group on Windows Server 2008;
* [SeSyncAgentPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is not assigned to any subject;
* [SeEnableDelegationPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is not assigned to any subject on Windows Vista and Server 2008;
  + By default, it is assigned to the “Administrators” group on Windows Server 2008 domain controllers;
* [SeTrustedCredManAccessPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system;
* [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK):
  + By default, it is assigned to local system.

The following is the list of Windows OS system access that may be assigned to a user account or a group in a local Windows OS machine:

* [SeInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx):
  + By default, it is assigned to the “Administrators” group, the “Backup Operators” group, the “Users” group on Windows Vista and Server 2008;
  + By default, it is assigned to the “Administrators” group, the “Backup Operators” group, the “Print Operators” group, the “Server Operators.” Group on Windows Server 2008 domain controllers;
* [SeDenyInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx):
  + By default, it is assigned to the local built-in “Guest” user account on Windows Vista;
  + By default, it is not assigned to any subject on Windows Server 2008;
* [SeRemoteInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx):
  + By default, it is assigned to the “Administrators” group, the “Remote Desktop Users” group on Windows Vista and Server 2008;
  + By default, it is assigned to the “Administrators” group on Windows Server 2008 domain controllers;
* [SeDenyRemoteInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx):
  + By default, it is not assigned to any subject;
* [SeNetworkLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx):
  + By default, it is assigned to the “Administrators” group, the “Backup Operators” group, the “Users” group, the “Everyone” (i.e. World) group on Windows Vista and Server 2008;
  + By default, it is assigned to the “Administrators” group, the “Authenticated Users” group, the “Enterprise Domain Controllers” group, the “Everyone” (i.e. World) group, the “Pre-Windows 2000 Compatible Access” group on Windows Server 2008 domain controllers;
* [SeDenyNetworkLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx):
  + By default, it is assigned to the local built-in “Guest” user account on Windows Vista;
  + By default, it is not assigned to any subject on Windows Server 2008;
* [SeBatchLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx):
  + By default, it is assigned to the “Administrators” group, the “Backup Operators” group;
* [SeDenyBatchLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx):
  + By default, it is not assigned to any subject;
* [SeServiceLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx):
  + By default, it is assigned to network service;
* [SeDenyServiceLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx):
  + By default, it is not assigned to any subject.

Also, as explained in the “Session ID assignment to an access token” section of this paper, when the Windows OS Authentication Service requests the Windows OS security reference monitor (SRM) to create an access token, it needs to assign a Session ID to the access token. If the registered logon process or the Windows OS service requesting the token creation belongs to Session X, then Session X is the Session ID assigned to the access token. For example, if Session X Window Logon State Maintaining Service requests the Windows OS Authentication Service for a primary token to represent the interactive logon user, then Session X is assigned as the Session ID of the primary token.

Consequently, this Commercial Grade OS Requirement Set “3.1.1.8” requirement is met.

## Addressing 3.1.1.9 “The OS shall protect authentication data from disclosure through the use of operating system provided cryptographic services”

This requirement is addressed by the Windows OS as follows, from the perspective of the lifecycle of a password after an administrator has created a user account requiring that the user of the user account “[must change password at next logon](http://msdn.microsoft.com/en-us/library/aa746542(VS.85).aspx)”.

### Password lifecycle

This section discusses the following stages in the password lifecycle.

#### Changing password

While logging on interactively, the user enters his/her old and new passwords in the dialog, titled “Change a password”, as provided by the window logon user interface service (aka LogonUI.exe) on the secure display area of the local machine, after the user initiates the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager instance.

##### Collecting the passwords from the secure display area

Together with the name of the user account for which its password change is being attempted, the old and new passwords travel locally from LogonUI.exe to either the Kerberos or the NTLM security provider via the window logon state maintaining service (aka winlogon.exe) with the help of the [ChangeAccountPassword()](http://msdn.microsoft.com/en-us/library/aa374755(VS.85).aspx) function exported by the Negotiate security provider. The Negotiate security provider tries the Kerberos security provider first using the Kerberos security provider’s [ChangeAccountPassword()](http://msdn.microsoft.com/en-us/library/aa374755(VS.85).aspx) function. If the Kerberos security provider returns [STATUS\_NO\_LOGON\_SERVERS](http://msdn.microsoft.com/en-us/library/cc704588.aspx), it implies that the user account for which its password change is being attempted is not a domain wide user account defined in the Windows OS Active Directory. Consequently, the NTLM security provider is tried with the NTLM security provider’s [ChangeAccountPassword()](http://msdn.microsoft.com/en-us/library/aa374755(VS.85).aspx) function, expecting that the user account for which its password change is being attempted is defined locally in the Windows OS SAM. If NTLM security provider also returns [STATUS\_NO\_LOGON\_SERVERS](http://msdn.microsoft.com/en-us/library/cc704588.aspx), it means that the user account is ill-defined.

As mentioned earlier in the “Isolation of user-entered credential through encryption” section, LogonUI.exe encrypts the old and new passwords. In this encryption, LogonUI.exe uses [RtlEncryptMemory() with the RTL\_ENCRYPT\_OPTION\_SAME\_LOGON option](http://msdn.microsoft.com/en-us/library/aa387693(VS.85).aspx). To decrypt the old and new passwords, the Kerberos security provider and NTLM security provider use [RtlDecryptMemory() with the RTL\_ENCRYPT\_OPTION\_SAME\_LOGON option](http://msdn.microsoft.com/en-us/library/aa387692(VS.85).aspx) also. The actual encryption and decryption behind RtlEncryptMemory() and RtlDecryptMemory() are implemented in the Windows OS security driver, ksecdd.sys. The encryption algorithm used is either AES 128 or 3DES (i.e. TDEA), depending on the length boundary of the data being encrypted.

In the case of ksecdd.sys, the cryptographic services come from ksecdd.sys itself. As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista ksecdd.sys has received FIPS-140-2 (Cert # 891). Windows Server 2008 ksecdd.sys has received FIPS-140-2 (Cert # 1007).

##### Kerberos Change Password protocol initiator

In the case where the user account for which its password change is being attempted is a domain wide user account defined in the Windows OS Active Directory, the Kerberos security provider uses the [Kerberos Change Password protocol](http://msdn.microsoft.com/en-us/library/ms817919.aspx) to communicate, potentially over the network, with a Kerberos KDC residing in a Windows OS domain controller. The protocol uses the old password to derive a session key for obtaining a change password service ticket. The new password is embedded cryptographically in a change password request, accompanied by the change password service ticket. As mentioned in the “Crypto Systems of the Kerberos security provider” section, the Kerberos security package has a fixed preferred list of crypto systems, as shown in the internal “Load CSystems” function of cryptdll.dll. In a pure Windows Vista and Windows Server 2008 environment, it is expected that the csAESk256 crypto system (as explained in the “Crypto Systems of the Kerberos security provider” section) is used, as the csAESk256 crypto system is the preferred crypto system on both Windows Vista and Windows Server 2008.

In the case of cryptdll.dll, it is seen from its internal “Check CNG” function that cryptdll.dll loads bcrypt.dll and uses its cryptographic services. As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista bcrypt.dll has received FIPS-140-2 (Cert # 892). Windows Server 2008 bcrypt.dll has received FIPS-140-2 (Cert # 1008).

The administrator-allowed-only operation for resetting a domain-wide user account’s password is conducted in the same matter; namely, using the [Kerberos Change Password protocol](http://msdn.microsoft.com/en-us/library/ms817919.aspx) to communicate, potentially over the network, with a Kerberos KDC residing on a Windows OS domain controller. Therefore, the above description of the cryptographic service aspects for the new password applies to this situation also.

##### Kerberos Change Password protocol handler

After the receiving Kerberos KDC handles the [Kerberos Change Password protocol](http://msdn.microsoft.com/en-us/library/ms817919.aspx) message, and, if successfully, it updates the new password hashes and related credentials for the user account residing in the database of the Active Directory.

The Windows Server 2008 Kerberos KDC supports the csAESk256 crypto system (as explained in the “Crypto Systems of the Kerberos security provider” section) also. Similar to the Kerberos security provider, the Kerberos KDC also uses the cryptographic services of bcrypt.dll through cryptdll.dll when processing the Kerberos Change Password protocol protected with the csAESk256 crypto system.

Assuming that the user account satisfies the necessary password-related restrictions and the new password meets the necessary password complexity and history requirements, the new password is used to update the following attributes associated with the domain wide user account:

* [ATT\_UNICODE\_PWD](http://msdn.microsoft.com/en-us/library/ms680513(VS.85).aspx)
  + the attribute value being updated is the encrypted NTLM MD4 hash of the new password after the RC4 encryption (using the internal “PEK Encrypt” function) [according to the system key (syskey) policy](http://support.microsoft.com/kb/818200);
  + the RC4 key for the above attribute encryption resides in the hidden [ATT\_PEK\_LIST](http://msdn.microsoft.com/en-us/library/ms679109(VS.85).aspx) of the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) and it is RC4-encrypted using the master key of the syskey policy;
* [ATT\_NT\_PWD\_HISTORY](http://msdn.microsoft.com/en-us/library/ms679004(VS.85).aspx)
  + the attribute value being updated is an update of the historical list of the encrypted NTLM MD4 password hashes with the latest inclusion of the encrypted NTLM MD4 hash of the new password after the RC4 encryption (using the internal “PEK Encrypt” function) [according to the system key (syskey) policy](http://support.microsoft.com/kb/818200);
  + the RC4 key for the above attribute encryption resides in the hidden [ATT\_PEK\_LIST](http://msdn.microsoft.com/en-us/library/ms679109(VS.85).aspx) of the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) and it is RC4-encrypted using the master key of the syskey policy;
* [ATT\_SUPPLEMENTAL\_CREDENTIALS](http://msdn.microsoft.com/en-us/library/ms679920(VS.85).aspx)
  + with the assistance of the local Kerberos KDC, the attribute value being updated includes the various cryptographic hashes of the new password associated with each of the supported Kerberos crypto systems;
  + with the assistance of the local Digest security provider, the attribute value being updated includes the various MD5 hashes of the new password associated with each of the supported Digest username/Realm forms;
  + the attribute value is encrypted through RC4 (using the internal “PEK Encrypt” function) [according to the system key (syskey) policy](http://support.microsoft.com/kb/818200);
  + the RC4 key for the above attribute encryption resides in the hidden [ATT\_PEK\_LIST](http://msdn.microsoft.com/en-us/library/ms679109(VS.85).aspx) of the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) and it is RC4-encrypted using the master key of the syskey policy.

The RC4 encryption and decryption of the above Active Directory object attributes, due to the syskey policy, occur at the database layer of the Active Directory in every Windows OS domain controller. The implementation of the RC4 encryption and decryption is not provided by a Windows OS cryptographic module, but residing within the Active Directory (ntdsa.dll and ntdsai.dll). However, because of the availability of the Windows OS BitLocker™ components for supporting the Full volume encryption, the above attributes are ultimately encrypted. As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista BitLocker™ components have received FIPS-140-2 (Cert # 947) and Windows Server 2008 BitLocker™ components have received FIPS-140-2 (Cert # 1054). Consequently, the password based authentication data stored as Active Directory object attributes are encrypted through the use of operating system provided cryptographic services, which are compliant with the FIPS-140-2 requirements.

The administrator-allowed-only operation for resetting a domain-wide user account’s password is conducted in the same matter; namely, the receiving Kerberos KDC handling of the [Kerberos Change Password protocol](http://msdn.microsoft.com/en-us/library/ms817919.aspx) at a Windows OS domain controller. Therefore, the above description of the cryptographic service aspects for the new password applies to this situation also.

##### Kerberos security provider notifies the local Windows OS credential manger

After the Kerberos security provider receives a successful response from the Kerberos KDC through the [Kerberos Change Password protocol](http://msdn.microsoft.com/en-us/library/ms817919.aspx), it notifies the local [Windows OS credential manager](http://www.microsoft.com/technet/security/guidance/identitymanagement/idmanage/p2pass_1.mspx?mfr=true) about the new password. The local [Windows OS credential manager](http://www.microsoft.com/technet/security/guidance/identitymanagement/idmanage/p2pass_1.mspx?mfr=true) resides in the same Windows OS process as the Kerberos security provider. The local [Windows OS credential manager](http://www.microsoft.com/technet/security/guidance/identitymanagement/idmanage/p2pass_1.mspx?mfr=true) writes the new password into a persistent “credentials” file in the user (roaming) profile for the user account. This “credentials” file resides in either

* the “AppData\Local\Microsoft\Credentials” subdirectory, or
* the “AppData\Roaming\Microsoft\Credentials” subdirectory

under the root directory of the user (roaming) profile. Before the writing to the “credentials” file, the local [Windows OS credential manager](http://www.microsoft.com/technet/security/guidance/identitymanagement/idmanage/p2pass_1.mspx?mfr=true) uses the services of the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) to encrypt the new password. The [Windows OS DPAPI manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) uses the cryptographic services of the Windows OS Enhanced Cryptographic Provider (rsaenh.dll). The default algorithm for the encryption is 3DES (i.e. TDEA). As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista rsaenh.dll has received FIPS-140-2 (Cert # 893). Windows Server 2008 rsaenh.dll has received FIPS-140-2 (Cert # 1010).

The administrator-allowed-only operation for resetting a domain-wide user account’s password is conducted in the same matter; namely, the Kerberos security provider’s notifications for the local Windows OS credential manager. Therefore, the above description of the cryptographic service aspects for the new password applies to this situation also.

##### Kerberos security provider notifies the local Windows OS data protection API manger

After the Kerberos security provider receives a successful response from the Kerberos KDC through the [Kerberos Change Password protocol](http://msdn.microsoft.com/en-us/library/ms817919.aspx), it notifies the local [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) about the password change. The local [Windows OS DPAPI manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) resides in the same Windows OS process as the Kerberos security provider. The [Windows OS DPAPI manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) re-encrypts the user account’s master key encryption key with a PKCS#5 V2 derivation of the SHA-1 hash of the new password using 3DES (i.e. TDEA). The user account’s master key encryption key is used to encrypt DPAPI protected items for a subject acting on behalf of the user account. The user account’s encrypted master key encryption key is stored in a DPAPI file in

* the “AppData\Roaming\Microsoft\Protect\<User SID of the user account>” subdirectory

under the root directory of the user (roaming) profile. With the user account’s encrypted master key encryption key being available in the user account’s roaming profile, the actual master key encryption key is available through the use of the user’s account current password, without needing to communicate with a Windows OS domain controller’s DPAPI master key recovery service.

The implementation of the 3DES (i.e. TDEA) encryption for the user account’s master key encryption key is not provided by a Windows OS cryptographic module, but residing within the Windows OS DPAPI manager.

However, because of the availability of the Windows OS BitLocker™ components for supporting the Full volume encryption, the above user account’s master key encryption key is ultimately encrypted. As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista BitLocker™ components have received FIPS-140-2 (Cert # 947) and Windows Server 2008 BitLocker™ components have received FIPS-140-2 (Cert # 1054). Consequently, the user account’s master key encryption key is encrypted through the use of operating system provided cryptographic services, which are compliant with the FIPS-140-2 requirements.

The administrator-allowed-only operation for resetting a domain-wide user account’s password is conducted in the same matter; namely, the Kerberos security provider’s notifications for the local Windows OS DPAPI manager. Therefore, the above description of the cryptographic service aspects for the new password applies to this situation also.

##### Caching a verifier of the user account password

After the Kerberos security provider receives a successful response from the Kerberos KDC through the [Kerberos Change Password protocol](http://msdn.microsoft.com/en-us/library/ms817919.aspx), it notifies the local NTLM security provider (via [MsV1\_0ChangeCachedPassword](http://msdn.microsoft.com/en-us/library/aa378766(VS.85).aspx)) to change the cached verifier of the user account’s valid password (namely a PKCS#5 V2 hash of the NTLM MD4 hash of the password), which may be used to authenticate a user attempting to logon to the user account locally when a Kerberos KDC in the local machine’s Windows OS domain cannot be contacted. The local NTLM security provider resides in the same Windows OS process as the Kerberos security provider. The cached verifier is encrypted using AES 128 with a persistent random key, which is stored locally as a named (“NL$KM”) LSA policy secret object. The encrypted cached verifier is then stored locally as a Windows OS registry key value under the “HKEY\_LOCAL\_MACHINE\SECURITY\Cache” registry key.

The AES 128 encryption of the cached verifier uses the cryptographic services of bcrypt.dll through the library functions of cryptdll.dll. Windows Vista bcrypt.dll has received FIPS-140-2 (Cert # 892). Windows Server 2008 bcrypt.dll has received FIPS-140-2 (Cert # 1008).

LSA policy secret objects are encrypted also. Recall that the Windows OS Authentication Service (aka lsass.exe) is the hosting Windows OS process of LSA policy secret objects. The Windows OS Authentication Service (aka lsass.exe) uses the master key of the syskey policy to decrypt/encrypt the value of the “PolEKList” LSA database attribute. The encryption algorithm of this decryption/encryption is AES 256. The decrypted value of the “PolEKList” attribute is the LSA database attribute encryption key. This LSA database attribute encryption key is used to decrypt/encrypt LSA policy secret objects, including the named (“NL$KM”) LSA policy secret object. The encryption algorithm of this decryption/encryption is also AES 256. Therefore, the persistent random key for decrypting/encrypting the cached verifiers of a user account’s valid password is also encrypted.

The implementation of the AES 256 encryption and decryption for the LSA policy secret objects and the LSA database attribute encryption key is not provided by a Windows OS cryptographic module, but residing within the Windows OS Authentication Service (aka lsass.exe). However, because of the availability of the Windows OS BitLocker™ components for supporting the Full volume encryption, the above attributes are ultimately encrypted. As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista BitLocker™ components have received FIPS-140-2 (Cert # 947) and Windows Server 2008 BitLocker™ components have received FIPS-140-2 (Cert # 1054). Consequently, the password based authentication data stored as LSA database attributes are encrypted through the use of operating system provided cryptographic services, which are compliant with the FIPS-140-2 requirements.

The administrator-allowed-only operation for resetting a domain-wide user account’s password is conducted in the same matter; namely, the Kerberos security provider’s notifications for the local NTLM security provider. Therefore, the above description of the cryptographic service aspects for the new password applies to this situation also.

##### Password changing for a user account defined in the local Windows OS SAM

In the case where the user account, for which its password change is being attempted, is defined locally in the Windows OS SAM, the NTLM security provider contacts the local SAM, residing in the same Windows OS process, via the [SamrUnicodeChangePasswordUser2()](http://msdn.microsoft.com/en-us/library/cc245708.aspx) interface. When using the [SamrUnicodeChangePasswordUser2()](http://msdn.microsoft.com/en-us/library/cc245708.aspx) interface, the NTLM security provider

* RC4-encrypts the new password using the key as being the NTLM MD4 hash of the old password as supplied the user;
* RC4-encrypts the NTLM MD4 hash of the old password, as supplied the user, using the key as being the NTLM MD4 hash of the new password as supplied the user.

The local SAM handles the [SamrUnicodeChangePasswordUser2()](http://msdn.microsoft.com/en-us/library/cc245708.aspx) interface. The local SAM uses its stored NTLM MD4 hash of the old password from the SAM equivalent of the ATT\_UNICODE\_PWD attribute for the user account. It

* RC4-decrypts the new password using the key as being its stored NTLM MD4 hash of the old password;
* RC4-decrypts the NTLM MD4 hash of the user-supplied old password, using the key as being the NTLM MD4 hash of the above decrypted new password;
* compares the decrypted NTLM MD4 hash of the user-supplied old password with its stored NTLM MD4 hash of the old password.

Assuming that

* the user-supplied NTLM MD4 hash of the old password and the stored NTLM MD4 hash of the old password match;
* the user account satisfies the necessary password-related restrictions;
* the new password meets the necessary password complexity and history requirements,

the local SAM then uses the new password to update the SAM equivalent of the following attributes associated with the local Windows OS SAM-defined user account:

* [ATT\_UNICODE\_PWD](http://msdn.microsoft.com/en-us/library/ms680513(VS.85).aspx)
  + the attribute value being updated is the encrypted NTLM MD4 hash of the new password after the RC4 encryption (using the internal “Encrypt Secret Data” function) [according to the system key (syskey) policy](http://support.microsoft.com/kb/818200);
  + the RC4 key for the above attribute encryption resides in the “Domain Key Information” element of the local Windows OS SAM account domain and it is RC4-encrypted using the master key of the syskey policy;
* [ATT\_NT\_PWD\_HISTORY](http://msdn.microsoft.com/en-us/library/ms679004(VS.85).aspx)
  + the attribute value being updated is an update of the historical list of the encrypted NTLM MD4 password hashes with the latest inclusion of the encrypted NTLM MD4 hash of the new password after the RC4 encryption (using the internal “Encrypt Secret Data” function) [according to the system key (syskey) policy](http://support.microsoft.com/kb/818200);
  + the RC4 key for the above attribute encryption resides in the “Domain Key Information” element of the local Windows OS SAM account domain and it is RC4-encrypted using the master key of the syskey policy.

The RC4 encryption and decryption of the above Windows OS SAM-defined user account attributes, due to the syskey policy, occur at the database layer of the Windows OS SAM in every Windows OS machine. The implementation of the RC4 encryption and decryption is not provided by a Windows OS cryptographic module, but residing within the Windows OS SAM (smasrv.dll). However, because of the availability of the Windows OS BitLocker™ components for supporting the Full volume encryption, the above attributes are ultimately encrypted. As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista BitLocker™ components have received FIPS-140-2 (Cert # 947) and Windows Server 2008 BitLocker™ components have received FIPS-140-2 (Cert # 1054). Consequently, the password based authentication data stored as Windows OS SAM-defined user account attributes are encrypted through the use of operating system provided cryptographic services, which are compliant with the FIPS-140-2 requirements.

The administrator-allowed-only operation for resetting a local Windows OS SAM user account’s password is conducted in the same matter; namely, the communication with the local Windows OS SAM over its local RPC interfaces. Therefore, the above description of the cryptographic service aspects for the new password applies to this situation also.

##### NTLM security provider notifies the local Windows OS credential manger

After the NTLM security provider receives a successful response from the local Windows OS SAM through the [SamrUnicodeChangePasswordUser2()](http://msdn.microsoft.com/en-us/library/cc245708.aspx) interface, it notifies the local [Windows OS credential manager](http://www.microsoft.com/technet/security/guidance/identitymanagement/idmanage/p2pass_1.mspx?mfr=true) about the new password. The local [Windows OS credential manager](http://www.microsoft.com/technet/security/guidance/identitymanagement/idmanage/p2pass_1.mspx?mfr=true) writes the new password into a persistent “credentials” file in the user profile for the user account. This “credentials” file resides in

* the “AppData\Local\Microsoft\Credentials” subdirectory

under the root directory of the user profile. Before the writing to the “credentials” file, the local [Windows OS credential manager](http://www.microsoft.com/technet/security/guidance/identitymanagement/idmanage/p2pass_1.mspx?mfr=true) uses the services of the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) to encrypt the new password. The [Windows OS DPAPI manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) uses the cryptographic services of the Windows OS Enhanced Cryptographic Provider (rsaenh.dll). The default algorithm for the encryption is 3DES (i.e. TDEA). As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista rsaenh.dll has received FIPS-140-2 (Cert # 893). Windows Server 2008 rsaenh.dll has received FIPS-140-2 (Cert # 1010).

The administrator-allowed-only operation for resetting a domain-wide user account’s password is conducted in the same matter; namely, the NTLM security provider’s notifications for the local Windows OS credential manager. Therefore, the above description of the cryptographic service aspects for the new password applies to this situation also.

##### NTLM security provider notifies the local Windows OS data protection API manger

After the NTLM security provider receives a successful response from the local Windows OS SAM through the [SamrUnicodeChangePasswordUser2()](http://msdn.microsoft.com/en-us/library/cc245708.aspx) interface, it notifies the local [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) about the password change with the old and new passwords.

A credential history item is a 3DES-encryption (i.e. TDEA-encryption) of the combined string of

* the SHA-1 hash of the old password;
* the NTLM MD4 hash of the old password

using the SHA-1 hash of the new password as the encryption key.

A credential history is a list of credential history items, which are ordered from oldest to the newest.

After creating the newest credential history item using the old and new passwords for the current password change, the local [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) appends it to the current credential history. The DPAPI manager writes the appended credential history to the “CREDHIST” file residing in

* the “AppData\Roaming\Microsoft\Protect\<User SID of the user account>” subdirectory

under the root directory of the user profile.

The [Windows OS DPAPI manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) re-encrypts the user account’s master key encryption key with a PKCS#5 V2 derivation of the SHA-1 hash of the new password using 3DES (i.e. TDEA). The user account’s master key encryption key is used to encrypt DPAPI protected items for a subject acting on behalf of the user account. The user account’s encrypted master key encryption key is stored in a DPAPI file in

* the “AppData\Roaming\Microsoft\Protect\<User SID of the user account>” subdirectory

under the root directory of the user (roaming) profile.

The [Windows OS DPAPI manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) RSA-encrypts the new password with the user account’s password recovery 2048-bit public key. The [Windows OS DPAPI manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) generates a cryptographically verifiable signature. This signature is a SHA HMAC of the encoded certificate for the password recovery 2048-bit public key, using a PKCS#5 V2 hash of the SHA-1 hash of the new password as the HMAC key. The signature can be used to verify the binding association between the password recovery 2048-bit public key and the new password. The RSA-encrypted password and the signature are then stored locally in a Windows OS registry key value under the “HKEY\_LOCAL\_MACHINE\SECURITY\Recovery\<user sid>” registry key. The password recovery 2048-bit public key certificate used to encrypt the new password must have a valid signature that has been validated by the [Windows OS DPAPI manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) using the old password. The password recovery 2048-bit public key and its corresponding private key would have been generated previously for the user account when the account’s owner exercised the “[Forgotten Password Wizard](http://support.microsoft.com/kb/930381)” using the old password. The password recovery private key would also have been saved to a removable password key disk according to the user manual of the “[Forgotten Password Wizard](http://support.microsoft.com/kb/930381)”. As explained in the user manual, it is important for the user account’s owner to store the disk in a safe location.

The implementation of the 3DES (i.e. TDEA) encryption for the user account’s credential history item and the user account’s master key encryption key is not provided by a Windows OS cryptographic module, but residing within the [Windows OS DPAPI manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx).

However, because of the availability of the Windows OS BitLocker™ components for supporting the Full volume encryption, the above user account’s credential history item and the user account’s master key encryption key are ultimately encrypted. As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista BitLocker™ components have received FIPS-140-2 (Cert # 947) and Windows Server 2008 BitLocker™ components have received FIPS-140-2 (Cert # 1054). Consequently, the user account’s credential history item and the user account’s master key encryption key are encrypted through the use of operating system provided cryptographic services, which are compliant with the FIPS-140-2 requirements.

The RSA-encryption of the new password is conducted using the Crypto API [CryptEncrypt()](http://msdn.microsoft.com/en-us/library/aa379924.aspx) of [crypt32.dll](http://msdn.microsoft.com/en-us/library/aa379884(VS.85).aspx). The Crypto API [CryptEncrypt()](http://msdn.microsoft.com/en-us/library/aa379924.aspx) uses the cryptographic services of the cryptographic services of the Windows OS Enhanced Cryptographic Provider (rsaenh.dll). As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista rsaenh.dll has received FIPS-140-2 (Cert # 893). Windows Server 2008 rsaenh.dll has received FIPS-140-2 (Cert # 1010).

The administrator-allowed-only operation for resetting a domain-wide user account’s password is conducted in the same matter; namely, the NTLM security provider’s notifications for the local Windows OS DPAPI manager. Therefore, the above description of the cryptographic service aspects for the new password applies to this situation also.

##### Remote password change for a user account defined in Windows OS SAM over Windows OS SAM RPC interfaces

The above has given a description of the password change operation through the dialog, titled “Change a password”, as provided by the window logon user interface service (aka LogonUI.exe). This is the recommended mode of operation for changing a user account’s password.

However, the Windows OS SAM also provides the following RPC interfaces for a user to remotely change the password for one of its user accounts:

* [SamrChangePasswordUser()](http://msdn.microsoft.com/en-us/library/cc211972.aspx)
  + this interface is moot for the case of LM OWF password because the security option of “[Network security: Do not store LAN Manager hash value on next password change](http://technet2.microsoft.com/windowsserver/en/library/393fa32d-04dd-4a15-b23d-3fc2b85588821033.mspx?mfr=true)” is enabled by default;
* [SamrOemChangePasswordUser2()](http://msdn.microsoft.com/en-us/library/cc211973.aspx)
  + this interface is moot because the security option of “[Network security: Do not store LAN Manager hash value on next password change](http://technet2.microsoft.com/windowsserver/en/library/393fa32d-04dd-4a15-b23d-3fc2b85588821033.mspx?mfr=true)” is enabled by default;
* [SamrUnicodeChangePasswordUser2()](http://msdn.microsoft.com/en-us/library/cc245708.aspx);
* [SamrUnicodeChangePasswordUser3()](http://msdn.microsoft.com/en-us/library/cc211975.aspx).

The protection of the passwords being transported over these interfaces across the network is provided through the following scheme.

* The remote client (of the interfaces) RC4-encrypts the new password using the key as being the NTLM MD4 hash of the old password.
* The remote client (of the interfaces) RC4-encrypts the NTLM MD4 hash of the old password using the key as being the NTLM MD4 hash of the new password as supplied the user.

The local SAM handles the interfaces as follows. The local SAM uses its stored NTLM MD4 hash of the old password from the SAM equivalent of the ATT\_UNICODE\_PWD attribute for the user account. It

* RC4-decrypts the new password using the key as being its stored NTLM MD4 hash of the old password;
* RC4-decrypts the NTLM MD4 hash of the client-supplied old password, using the key as being the NTLM MD4 hash of the above decrypted new password;
* compares the decrypted NTLM MD4 hash of the client-supplied old password with its stored NTLM MD4 hash of the old password.

If the above protection scheme of the passwords being transported over the above RPC interfaces across the network is deemed as inadequate, then the remote changing of password for a Windows OS SAM user account is not recommended. Additionally, there is the non-default “EnforceFIPSRequirementForPasswordOperations” policy which can be applied on the client side of the above RPC interfaces in the “samlib.dll” code library DLL. When enabled, this policy disallows a caller subject of samlib.dll to exercise the above Windows OS SAM RPC interfaces to contact a remote Windows OS machine. The policy may be enabled by an administrator through the “EnforceFIPSRequirementForPasswordOperations” Windows OS registry key value under the following registry key.

* “HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa”

#### Submitting a password for user authentication

While logging on interactively, the user enters his/her password in the “Log on to: <domain name>” dialog, provided by the window logon user interface service (aka LogonUI.exe) on the secure display area of the local machine, after the user initiates the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager instance.

##### Collecting the password from the secure display area

Together with the name of the user account for which its logging on is being attempted, the password travels locally from LogonUI.exe to either the Kerberos or the NTLM security provider via the window logon state maintaining service (aka winlogon.exe) with the help of the [LsaLogonUser()](http://msdn.microsoft.com/en-us/library/aa378292(VS.85).aspx) function exported by the Negotiate security provider. The Negotiate security provider tries the Kerberos security provider first using the Kerberos security provider’s  [LogonUserEx2()](http://msdn.microsoft.com/en-us/library/aa380114(VS.85).aspx) function. If the Kerberos security provider returns [STATUS\_NO\_LOGON\_SERVERS](http://msdn.microsoft.com/en-us/library/cc704588.aspx) or similar, it implies that either

* the user account for which its logging on is being attempted is not a domain wide user account defined in the Windows OS Active Directory;
* there is no network connection to contact any Windows OS domain controller in the Windows OS domain of user account.

Consequently, the NTLM security provider is tried with the NTLM security provider’s  [LogonUserEx2()](http://msdn.microsoft.com/en-us/library/aa380114(VS.85).aspx) function, expecting that either

* the user account for which its logging on is being attempted is defined locally in the Windows OS SAM;
* the user account may have a cached password verifier residing locally in the “HKEY\_LOCAL\_MACHINE\SECURITY\Cache” registry key.

If the NTLM security provider also returns [STATUS\_NO\_LOGON\_SERVERS](http://msdn.microsoft.com/en-us/library/cc704588.aspx), it means that the user account is ill-defined.

As mentioned earlier in the “Isolation of user-entered credential through encryption” section, LogonUI.exe encrypts the password. In this encryption, LogonUI.exe uses [RtlEncryptMemory() with the RTL\_ENCRYPT\_OPTION\_SAME\_LOGON option](http://msdn.microsoft.com/en-us/library/aa387693(VS.85).aspx). To decrypt the password, the Kerberos security provider and NTLM security provider use [RtlDecryptMemory() with the RTL\_ENCRYPT\_OPTION\_SAME\_LOGON option](http://msdn.microsoft.com/en-us/library/aa387692(VS.85).aspx) also. The actual encryption and decryption behind RtlEncryptMemory() and RtlDecryptMemory() are implemented in the Windows OS security driver, ksecdd.sys. The encryption algorithm used is either AES 128 or 3DES (i.e. TDEA), depending on the length boundary of the data being encrypted.

In the case of ksecdd.sys, the cryptographic services come from ksecdd.sys itself. As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista ksecdd.sys has received FIPS-140-2 (Cert # 891). Windows Server 2008 ksecdd.sys has received FIPS-140-2 (Cert # 1007).

##### Kerberos Authentication Service Request

In the case where the user account for which its logging on is being attempted is a domain wide user account defined in the Windows OS Active Directory, the Kerberos security provider creates a [Kerberos Authentication Service Request (KRB\_AS\_REQ)](http://msdn.microsoft.com/en-us/library/aa374739(VS.85).aspx) to communicate, potentially over the network, with a Kerberos KDC residing on a Windows OS domain controller. The protocol for KRB\_AS\_REQ uses the password to derive a session key for obtaining a ticket granting ticket from the Kerberos KDC’s ticket-granting service. As mentioned in the “Crypto Systems of the Kerberos security provider” section, the Kerberos security package has a fixed preferred list of crypto systems, as shown in the internal “Load CSystems” function of cryptdll.dll. In a pure Windows Vista and Windows Server 2008 environment, it is expected that the csAESk256 crypto system (as explained in the “Crypto Systems of the Kerberos security provider” section) is used, as the csAESk256 crypto system is the preferred crypto system on both Windows Vista and Windows Server 2008.

In the case of cryptdll.dll, it is seen from its internal “Check CNG” function that cryptdll.dll loads bcrypt.dll and uses its cryptographic services. As shown on <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/1401val2008.htm>, Windows Vista bcrypt.dll has received FIPS-140-2 (Cert # 892). Windows Server 2008 bcrypt.dll has received FIPS-140-2 (Cert # 1008).

##### Kerberos Authentication Service Request handler

The receiving Kerberos KDC handles the [Kerberos Authentication Service Request (KRB\_AS\_REQ)](http://msdn.microsoft.com/en-us/library/aa374739(VS.85).aspx) message. In its processing of the message, the Kerberos KDC uses the cryptographic password hash of the corresponding Kerberos crypto system in the [ATT\_SUPPLEMENTAL\_CREDENTIALS](http://msdn.microsoft.com/en-us/library/ms679920(VS.85).aspx) of the user account, for which its logging on is being attempted.

Assuming that the KRB\_AS\_REQ message processing is successful and the user account satisfies the necessary password-related and authentication-related restrictions, the user account is considered as “logged on” to the Windows OS domain, at least from the Windows OS domain controller’s perspective. The Kerberos KDC then returns a Kerberos Authentication Service Reply ([KRB\_AS\_REP](http://msdn.microsoft.com/en-us/library/aa374739(VS.85).aspx)) message back to the originating Kerberos security provider to indicate the “logged on” status to the Windows OS domain. After receiving [KRB\_AS\_REP](http://msdn.microsoft.com/en-us/library/aa374739(VS.85).aspx) message, the Kerberos security provider continues the remaining of the user account logging on processing locally.

The Windows Server 2008 Kerberos KDC supports the csAESk256 crypto system (as explained in the “Crypto Systems of the Kerberos security provider” section) also. Similar to the Kerberos security provider, the Kerberos KDC also uses the cryptographic services of bcrypt.dll through cryptdll.dll when processing the Kerberos Change Password protocol protected with the csAESk256 crypto system.

##### Logging on attempt through a cached password verifier

In the case where the user account for which its logging on is being attempted is a domain wide user account defined in the Windows OS Active Directory, but no Kerberos KDCs can be contacted, the NTLM security provider attempts the user account logging on through the use of a cached password verifier residing in the registry key value under the “HKEY\_LOCAL\_MACHINE\SECURITY\Cache” registry key. The NTLM security provider uses the user supplied password to compute a verifier. If the computed verifier matches the cached verifier, then the user supplied password for logging to the specific user account is deemed as validated. Subsequently, the NTLM security provider continues the remaining of the user account logging on processing locally.

##### Local attempt to log on to a user account defined in the local Windows OS SAM

In the case where the user account for which its logging on is being attempted is defined locally in the Windows OS SAM, the NTLM security provider attempts the user account logging on through the use of the local SAM’s stored NTLM MD4 password hash from the SAM equivalent of the [ATT\_UNICODE\_PWD](http://msdn.microsoft.com/en-us/library/ms680513(VS.85).aspx) attribute for the user account. The NTLM security provider also uses the user supplied password to compute a NTLM MD4 password hash. If the computed NTLM MD4 password hash matches the local SAM’s stored NTLM MD4 password hash, then the user supplied password for logging to the specific user account is deemed as validated. Subsequently, the NTLM security provider continues the remaining of the user account logging on processing locally.

The above has explained the lifecycle of a password and it has described where cryptographic services are applied within the lifecycle. As a result, this Commercial Grade OS Requirement Set “3.1.1.9” requirement is addressed.

## Addressing 3.1.1.10 “The OS shall provide an administrator role that is separate from untrusted users”

This requirement is addressed by the Windows OS as follows.

The Windows OS defines the following relative IDs of group SIDs for an administrative user account to possess. The resultant group SIDs are distinct from other SIDs. Therefore, the use of the groups SIDs to assign

* access control ACEs to groups represented by the group SIDs in a security descriptor for an object or a system functional interface; or
* Windows OS privileges to groups represented by the group SIDs in a system functional interface

separates the administrative capabilities from the normal capabilities which are available only to standard users.

* [DOMAIN\_USER\_RID\_ADMIN](http://msdn.microsoft.com/en-us/library/aa379649.aspx);
* [DOMAIN\_GROUP\_RID\_ADMINS](http://msdn.microsoft.com/en-us/library/aa379649.aspx);
* [DOMAIN\_ALIAS\_RID\_ADMINS](http://msdn.microsoft.com/en-us/library/aa379649.aspx);
* [DOMAIN\_GROUP\_RID\_ENTERPRISE\_ADMINS](http://msdn.microsoft.com/en-us/library/aa379649.aspx).

Consequently, this Commercial Grade OS Requirement Set “3.1.1.10” requirement is satisfied.

## Addressing 3.1.1.11 “The OS shall provide the authorized administrator the ability to set user attributes”

This requirement is addressed by the Windows OS as follows.

The Windows OS defines user account objects in the Active Directory (for Windows OS domain wide user accounts) and in the SAM of an individual machine (for Windows OS local user accounts on the machine). In either case, the following lists a subset of attributes that may be set or configured by an administrator:

* [ATT\_NT\_SECURITY\_DESCRIPTOR](http://msdn.microsoft.com/en-us/library/ms679006(VS.85).aspx) (if the user account is domain based) or the security descriptor for the user account (if the user account is defined on a local machine)
  + It defines the access control security descriptor for gaining access to the user account attributes;
  + It also specifies the auditing requirement when an access to the attributes occurs;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_SCRIPT\_PATH](http://msdn.microsoft.com/en-us/library/ms679656(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It stores the path to the [logon script](http://technet2.microsoft.com/windowsserver/en/library/12cedcb6-a076-461d-bc73-0cc4048eff0d1033.mspx?mfr=true) for the user account so that certain policy elements of the user account are enforced, through the execution of the script, on the logon machine after the logging on to the user account;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_PROFILE\_PATH](http://msdn.microsoft.com/en-us/library/ms679422(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It stores the path to the [user profile](http://technet2.microsoft.com/windowsserver/en/library/093238f3-5064-470e-a281-0eb1c28b9cf01033.mspx?mfr=true) for the user account so that certain policy elements of the user account are enforced, through the configurations made in the profile, on the logon machine after the logging on to the user account;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It specifies the [hours](http://technet2.microsoft.com/windowsserver/en/library/a0f007ba-b3ef-4c1f-8836-9087fe5eb08d1033.mspx?mfr=true) that the user account is allowed to be logged on in the Windows OS domain;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_IS\_MEMBER\_OF\_DL](http://msdn.microsoft.com/en-us/library/ms677099(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It contains the distinguished names of the groups to which this user account directly belongs to;
  + Group nesting is possible in Windows OS;
  + A [group object](http://msdn.microsoft.com/en-us/library/ms682251(VS.85).aspx) also has its own ATT\_IS\_MEMBER\_OF\_DL attribute and so SAM is responsible to conduct the group membership expansion;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_UNICODE\_PWD](http://msdn.microsoft.com/en-us/library/ms680513(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It is not readable from outside the Windows OS process for SAM and it is not writable by a subject directly;
  + It stores the NTLM password hash in the format suitable for the user authentication being conducted by the NTLM security provider (aka msv1\_0 security provider);
  + If the user account is domain based, it can be updated by a subject with the proven knowledge of the current password, in the case where there are the ACCESS\_ALLOWED\_OBJECT\_ACE ACEs and not the ACCESS\_DENIED\_OBJECT\_ACE ACEs for the GUID\_CONTROL\_UserChangePassword property set to allow and not deny everyone and the principal self in the user account object’s security descriptor;
  + If the user account is locally defined, it can be updated by a subject with the proven knowledge of the current password, in the case where there are ACCESS\_ALLOWED\_ACE ACEs for USER\_CHANGE\_PASSWORD to allows everyone and the principal self in the user account object’s security descriptor;
  + If the user account is domain based, it can be updated by an administrator without the knowledge of the current password as there is the ACCESS\_ALLOWED\_OBJECT\_ACE ACE for the GUID\_CONTROL\_UserForceChangePassword property set to allow the administrator in the user account object’s default security descriptor;
  + If the user account is locally defined, it can be updated by an administrator without the knowledge of the current password as there is the ACCESS\_ALLOWED\_ACE ACE for USER\_FORCE\_PASSWORD\_CHANGE to allow the administrator in the user account object’s default security descriptor;
  + It is encrypted when residing in the database of SAM or of Active Directory [according to the system key (syskey) policy](http://support.microsoft.com/kb/818200);
* [ATT\_USER\_ACCOUNT\_CONTROL](http://msdn.microsoft.com/en-us/library/ms680832(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + Among others, it contains the USER\_ACCOUNT\_AUTO\_LOCKED, USER\_ACCOUNT\_DISABLED code values for supporting the account lock out policy described earlier;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_MS\_DS\_CACHED\_MEMBERSHIP](http://msdn.microsoft.com/en-us/library/ms677402(VS.85).aspx) (if the user account is domain based)
  + It stores the cached group memberships of the user account while still within the [cached membership staleness time](http://technet2.microsoft.com/WindowsServer/en/Library/0d34c3b9-499b-41d3-a55f-527ce61e78581033.mspx?mfr=true);
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_MS\_DS\_ALLOWED\_TO\_DELEGATE\_TO](http://msdn.microsoft.com/en-us/library/ms677183(VS.85).aspx) (if the user account is domain based)
  + It is relevant only if the user account represents an identity to run a service;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_SERVICE\_PRINCIPAL\_NAME](http://msdn.microsoft.com/en-us/library/ms679785(VS.85).aspx) (if the user account is domain based)
  + It is relevant only if the user account represents an identity to run a service;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_MS\_DS\_SUPPORTED\_ENCRYPTION\_TYPES](http://msdn.microsoft.com/en-us/library/ms677827(VS.85).aspx) (if the user account is domain based);
  + It specifies the encryption algorithms supported by the user account;
  + The Kerberos KDC uses the information in this attribute while generating a service ticket for the user account;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) (if the user account is domain based)
  + It contains associated mappings for X.509 public key certificates or external Kerberos user accounts to the user account for the purpose of public key certificate or smart card based user authentication;
  + By default, a standard user does not possess the right to update this attribute value.

Similarly, the access control rights (such as GROUP\_ADD\_MEMBER or the RIGHT\_DS\_WRITE\_PROPERTY associated with the GUID\_A\_MEMBER property) for an administrator are specified in the security descriptor of a Windows OS defined user account object or group object in the SAM of an individual machine or in the Active Directory. The administrator may set or modify group memberships of a user account, and hence impacting on the user security attributes associated with the user account.

Finally, the access control rights (such as [ACCOUNT\_ADJUST\_PRIVILEGES](http://msdn.microsoft.com/en-us/library/ms721750(VS.85).aspx) or [ACCOUNT\_ADJUST\_SYSTEM\_ACCESS](http://msdn.microsoft.com/en-us/library/ms721750(VS.85).aspx)) for an administrator are specified in the security descriptor of a Windows OS LSA policy Account object in the Windows OS authentication service's policy database. The administrator may set or modify the Windows OS privileges or system access for assigning to a Windows OS defined user account or a group containing the user account, and hence impacting on the user security attributes associated with the user account.

Consequently, this Commercial Grade OS Requirement Set “3.1.1.11” requirement is satisfied.

## Addressing 3.1.1.12 “The OS shall re-authenticate the user when changing authentication data”

This requirement is addressed by the descriptions given in the “Changing password” section as part of the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.9” requirement.

## Addressing 3.1.1.13 “The OS shall provide a mechanism to verify that each user password when set meets authorized administrator configurable password characteristics”

This requirement is addressed by the Windows OS as follows.

The Windows OS supports a [password complexity domain policy (DOMAIN\_PASSWORD\_COMPLEX)](http://support.microsoft.com/kb/279890). This policy is enforced by the Windows OS Local Security Account Management (SAM) (samsrv.dll) in its internal “Check Strong Password Restrictions” function. When enabled, this policy requires that a new password must meet the following minimum requirements:

* the password must not contain the password owner’s user account name or parts of the name that exceed two consecutive characters;
* the password must have a length of at least six characters;
* the password must contains characters from at least 3 of the following 5 types:
* Decimal digit characters ([C1\_DIGIT](http://msdn.microsoft.com/en-us/library/cc194821.aspx));
* Uppercase characters ([C1\_UPPER](http://msdn.microsoft.com/en-us/library/cc194821.aspx));
* Lowercase characters ([C1\_LOWER](http://msdn.microsoft.com/en-us/library/cc194821.aspx));
* Linguistic characters ([C1\_ALPHA](http://msdn.microsoft.com/en-us/library/cc194821.aspx));
* Special characters:
  + a character in the following string“`~!@#$%^&\*\_-+=|\\{}[]:;\"'<>,.”.

By default, the password complexity policy is enabled on Windows OS domain controllers and is disabled on stand-alone Windows OS servers or workstations. Also, by default, member computers of a Windows OS domain follow the configuration of their Windows OS domain controllers.

In addition, the Windows OS supports the [minimum password length domain policy (MinPasswordLength)](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx). This policy is enforced by the Windows OS Local Security Account Management (SAM) (samsrv.dll) in its internal “Validate Check Password Restrictions” function. This policy requires that the length of a new password must be at least the value specified in the policy. By default, the minimum password length is zero.

Consequently, this Commercial Grade OS Requirement Set “3.1.1.13” requirement is satisfied.

## Addressing 3.1.1.14 “The OS shall automatically disable a user account when the current time has passed the authorized administrator specified time interval in which the user account has not been logged on successfully since the user account’s last successful logon time”

This requirement is addressed by the Windows OS as follows.

The Windows OS supports the domain wide “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” policy. Under this policy, a logon failure with the [STATUS\_PASSWORD\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) status is returned when a logging on to a specific user account is attempted in the situation, where the current time has passed the time specified in the policy since the last time when the password was reset successfully.

We believe that the domain wide “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” policy also meets the intent of this requirement, but without reducing a lot of the user friendliness. The user friendliness is maintained because the user owner of the user account does not need to contact an administrator out-of-band for having his/her user account re-enabled.

Assuming that the “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” is equal or lesser than the “time interval in which the user account has not been logged on successfully since the last successful logon time”, then the impact to the logging on processing for the user account is the same (when the logging on time has passed the “time interval in which the user account has not been logged on successfully since the last successful logon time”). Namely, the logging on attempt is a failure, due to [STATUS\_PASSWORD\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx).

On the other hand, having the “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” larger than the “time interval in which the user account has not been logged on successfully since the last successful logon time” seems to be a bad practice because it implies that an administrator wishes to disable a user account before the password of the user account actually passes its maximum age. In that case, granting the user the ability to change his/her password by himself/herself is moot. Obviously, such outcome should not be an administrator’s best intent.

As a result, this requirement is addressed due to the availability of the “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” policy. For the purpose of meeting this requirement, an administrator could treat the “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” as the “time interval in which the user account has not been logged on successfully since the last successful logon time”.

Actually, the “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” policy has an advantage over the use of the “time interval in which the user account has not been logged on successfully since the last successful logon time” to disable a user account. The “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” policy covers the situation of an interactive logon session which is inactive indefinitely or for a long period of time. Because of the need of the interactive session locking capabilities as specified in the Commercial Grade OS Requirement Set requirements “3.2.1.2”, “3.2.1.3” and “3.2.1.5”, the “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” policy is still enforced when the user attempts to unlock the user’s interactive session by entering the required user account’s password.

# Meeting the “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Management Requirements”

In the Commercial Grade OS Requirement Set, there are 7 individual management requirements under the heading of “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Management Requirements”. They are listed as “3.1.2.n”, where n = 1, 2, 3, 4, 5, 6, and 7.

## Addressing 3.1.2.1 “The OS shall allow only authorized administrators the ability to create and manage user accounts”

This requirement is addressed by the Windows OS as follows.

The Windows OS SAM has the [SamrCreateUserInDomain()](http://msdn.microsoft.com/en-us/library/cc245771.aspx) interface for an authorized subject to create a locally defined user account object in its account domain. This interface requires the caller subject to possess the [DOMAIN\_CREATE\_USER](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right. By default, this [DOMAIN\_CREATE\_USER](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right is granted to

* an administrator;
* an account operator.

The Windows OS LDAP server has the LDAP interface for an authorized subject to create a domain wide user account object as a child object in the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) of the Active Directory using the [LDAP\_ADD\_CMD](http://msdn.microsoft.com/en-us/library/aa366139(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_CREATE\_CHILD right in the security descriptor for the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx). By default, this RIGHT\_DS\_CREATE\_CHILD right is granted to an administrator.

Finally, the management aspect of user accounts through the updating of attributes defined for user account objects has been described in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement. In particular, the specific attributes that do not allow a standard user to update their values have been stated explicitly.

Consequently, this Commercial Grade OS Requirement Set “3.1.2.1” requirement is satisfied.

## Addressing 3.1.2.2 “The OS shall provide authorized administrators the ability to specify a time interval in which the user account has not been logged on successfully so that a user account is automatically disabled after the time interval has elapsed”

This requirement is addressed by the Windows OS as follows.

The Windows OS supports the domain wide “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” policy. Under this policy, a logon failure with the [STATUS\_PASSWORD\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) status is returned when a logging on to a specific user account is attempted in the situation, where the current time has passed the time specified in the policy since the last time when the password was reset successfully.

We believe that the administrator’s ability to set the domain wide “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” policy also meets the intent of this requirement, but without reducing a lot of the user friendliness. The user friendliness is maintained because the user of the user account does not need to contact an administrator out-of-band for having his/her user account re-enabled.

Assuming that the “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” is equal or lesser than the “time interval in which the user account has not been logged on successfully since the last successful logon time”, then the impact to the logging on processing for the user account is the same (when the logging on time has passed the “time interval in which the user account has not been logged on successfully since the last successful logon time”). Namely, the logging on attempt is a failure, due to [STATUS\_PASSWORD\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx).

On the other hand, having the “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” larger than the “time interval in which the user account has not been logged on successfully since the last successful logon time” seems to be a bad practice because it implies that an administrator wishes to disable a user account before the password of the user account actually passes its maximum age. In that case, granting the user the ability to change his/her password by himself/herself is moot. Obviously, such outcome should not be an administrator’s best intent.

As a result, this requirement is addressed due to the availability of the “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” policy. For the purpose of meeting this requirement, an administrator could treat the “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” as the “time interval in which the user account has not been logged on successfully since the last successful logon time”.

Actually, the “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” policy has an advantage over the use of the “time interval in which the user account has not been logged on successfully since the last successful logon time” to disable a user account. The “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” policy covers the situation of an interactive logon session which is inactive indefinitely or for a long period of time. Because of the need of the interactive session locking capabilities as specified in the Commercial Grade OS Requirement Set requirements “3.2.1.2”, “3.2.1.3” and “3.2.1.5”, the “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” policy is still enforced when the user attempts to unlock the user’s interactive session by entering the required user account’s password.

## Addressing 3.1.2.3 “The OS shall allow only authorized administrators the ability to initially set and modify user security attributes (other than authentication data)”

This requirement is addressed by the Windows OS as follows.

The management aspect of user accounts through the updating of attributes defined for user account objects has been described in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement and “3.1.1.11” requirement. In particular, the specific attributes that do not allow a standard user to update their values have been stated explicitly.

## Addressing 3.1.2.4 “The OS shall allow only authorized administrators the ability to initialize user authentication data”

This requirement is addressed by the Windows OS as follows.

As described in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement, for the [ATT\_UNICODE\_PWD](http://msdn.microsoft.com/en-us/library/ms680513(VS.85).aspx) attribute (if the user account is domain based) or its equivalent attribute for a locally defined user account, the Windows OS operates in the following default behavior:

* If the user account is domain based, it can be updated by an administrator without the knowledge of the current password as there is the ACCESS\_ALLOWED\_OBJECT\_ACE ACE for the GUID\_CONTROL\_UserForceChangePassword property set to allow the administrator in the user account object’s default security descriptor;
* If the user account is locally defined, it can be updated by an administrator without the knowledge of the current password as there is the ACCESS\_ALLOWED\_ACE ACE for USER\_FORCE\_PASSWORD\_CHANGE to allow the administrator in the user account object’s default security descriptor.

## Addressing 3.1.2.5 “The OS shall restrict the ability to modify authentication data to authorized administrators and users authorized to modify their own authentication data”

This requirement is addressed by the Windows OS as follows.

As described in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement, for the [ATT\_UNICODE\_PWD](http://msdn.microsoft.com/en-us/library/ms680513(VS.85).aspx) attribute (if the user account is domain based) or its equivalent attribute for a locally defined user account, the Windows OS operates in the following default behavior:

* If the user account is domain based, it can be updated by a subject with the proven knowledge of the current password, in the case where there are the ACCESS\_ALLOWED\_OBJECT\_ACE ACEs and not the ACCESS\_DENIED\_OBJECT\_ACE ACEs for the GUID\_CONTROL\_UserChangePassword property set to allow and not deny everyone and the principal self in the user account object’s security descriptor;
* If the user account is locally defined, it can be updated by a subject with the proven knowledge of the current password, in the case where there are ACCESS\_ALLOWED\_ACE ACEs for USER\_CHANGE\_PASSWORD to allows everyone and the principal self in the user account object’s security descriptor;
* If the user account is domain based, it can be updated by an administrator without the knowledge of the current password as there is the ACCESS\_ALLOWED\_OBJECT\_ACE ACE for the GUID\_CONTROL\_UserForceChangePassword property set to allow the administrator in the user account object’s default security descriptor;
* If the user account is locally defined, it can be updated by an administrator without the knowledge of the current password as there is the ACCESS\_ALLOWED\_ACE ACE for USER\_FORCE\_PASSWORD\_CHANGE to allow the administrator in the user account object’s default security descriptor.

## Addressing 3.1.2.6 “The OS shall allow only authorized administrators the ability to initialize and modify authentication mechanism attributes”

Application Note from the Commercial Grade OS Requirement Set: Authentication mechanism attributes include password length, password complexity, number of allowable unsuccessfully authentication attempts, password history, etc.

This requirement is addressed by the Windows OS as follows.

The Windows OS supports the enforcement of the following domain policies or attributes.

* “[minimum password length](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” domain policy (MinPasswordLength)
  + By default, the minimum password length is zero.
  + The Windows OS SAM has the interface of [SamrSetInformationDomain() for the DomainPasswordInformation DomainInformationClass](http://msdn.microsoft.com/en-us/library/cc245788.aspx) for an authorized subject to set the minimum password length domain policy of its account domain. This interface requires the caller subject to possess the [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right. By default, this [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right is granted to an administrator.
  + The Windows OS LDAP server has the LDAP interface for an authorized subject to set the minimum password length domain policy of the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) in the Active Directory using the [LDAP\_MODIFY\_CMD](http://msdn.microsoft.com/en-us/library/aa366940(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set in the security descriptor for the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx). By default, this RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set is granted to an administrator.
* “[password complexity](http://support.microsoft.com/kb/279890)” domain policy (DOMAIN\_PASSWORD\_COMPLEX)
  + By default, the password complexity domain policy is enabled on Windows OS domain controllers and is disabled on stand-alone Windows OS servers or workstations. Also, by default, member computers of a Windows OS domain follow the configuration of their Windows OS domain controllers.
  + The Windows OS SAM has the interface of [SamrSetInformationDomain() for the DomainPasswordInformation DomainInformationClass](http://msdn.microsoft.com/en-us/library/cc245788.aspx) for an authorized subject to set the password complexity domain policy of its account domain. This interface requires the caller subject to possess the [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right. By default, this [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right is granted to an administrator.
  + The Windows OS LDAP server has the LDAP interface for an authorized subject to set the password complexity domain policy of the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) in the Active Directory using the [LDAP\_MODIFY\_CMD](http://msdn.microsoft.com/en-us/library/aa366940(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set in the security descriptor for the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx). By default, this RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set is granted to an administrator.
* “[maximum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” domain policy (MaxPasswordAge)
  + By default, the maximum password age is zero.
  + The Windows OS SAM has the interface of [SamrSetInformationDomain() for the DomainPasswordInformation DomainInformationClass](http://msdn.microsoft.com/en-us/library/cc245788.aspx) for an authorized subject to set the maximum password age domain policy of its account domain. This interface requires the caller subject to possess the [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right. By default, this [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right is granted to an administrator.
  + The Windows OS LDAP server has the LDAP interface for an authorized subject to set the maximum password age domain policy of the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) in the Active Directory using the [LDAP\_MODIFY\_CMD](http://msdn.microsoft.com/en-us/library/aa366940(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set in the security descriptor for the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx). By default, this RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set is granted to an administrator.
* “[minimum password age](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” domain policy (MinPasswordAge)
  + By default, the minimum password age is zero.
  + The Windows OS SAM has the interface of [SamrSetInformationDomain() for the DomainPasswordInformation DomainInformationClass](http://msdn.microsoft.com/en-us/library/cc245788.aspx) for an authorized subject to set the minimum password age domain policy of its account domain. This interface requires the caller subject to possess the [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right. By default, this [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right is granted to an administrator.
  + The Windows OS LDAP server has the LDAP interface for an authorized subject to set the minimum password age domain policy of the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) in the Active Directory using the [LDAP\_MODIFY\_CMD](http://msdn.microsoft.com/en-us/library/aa366940(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set in the security descriptor for the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx). By default, this RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set is granted to an administrator.
* “[password history length](http://windowshelp.microsoft.com/Windows/en-US/Help/279d63a1-e0cd-46a1-92f5-09b944f80fa01033.mspx)” domain policy (PasswordHistoryLength);
  + By default, the password history length is zero.
  + The Windows OS SAM has the interface of [SamrSetInformationDomain() for the DomainPasswordInformation DomainInformationClass](http://msdn.microsoft.com/en-us/library/cc245788.aspx) for an authorized subject to set the password history length domain policy of its account domain. This interface requires the caller subject to possess the [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right. By default, this [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right is granted to an administrator.
  + The Windows OS LDAP server has the LDAP interface for an authorized subject to set the password history length domain policy of the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) in the Active Directory using the [LDAP\_MODIFY\_CMD](http://msdn.microsoft.com/en-us/library/aa366940(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set in the security descriptor for the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx). By default, this RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set is granted to an administrator.
* “[lockout duration](http://msdn.microsoft.com/en-us/library/cc232773.aspx)” domain policy (LockoutDuration)
  + By default, the lockout duration is 30 minutes.
  + The Windows OS SAM has the interface of [SamrSetInformationDomain() for the DomainLockoutInformation DomainInformationClass](http://msdn.microsoft.com/en-us/library/cc245788.aspx) for an authorized subject to set the lockout duration domain policy of its account domain. This interface requires the caller subject to possess the [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right. By default, this [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right is granted to an administrator.
  + The Windows OS LDAP server has the LDAP interface for an authorized subject to set the lockout duration domain policy of the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) in the Active Directory using the [LDAP\_MODIFY\_CMD](http://msdn.microsoft.com/en-us/library/aa366940(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set in the security descriptor for the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx). By default, this RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set is granted to an administrator.
* “[lockout threshold](http://msdn.microsoft.com/en-us/library/cc232773.aspx)” domain policy (LockoutThreshold);
  + By default, the lockout threshold is zero.
  + The Windows OS SAM has the interface of [SamrSetInformationDomain() for the DomainLockoutInformation DomainInformationClass](http://msdn.microsoft.com/en-us/library/cc245788.aspx) for an authorized subject to set the “lockout threshold” domain policy of its account domain. This interface requires the caller subject to possess the [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right. By default, this [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right is granted to an administrator.
  + The Windows OS LDAP server has the LDAP interface for an authorized subject to set the “lockout threshold” domain policy of the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) in the Active Directory using the [LDAP\_MODIFY\_CMD](http://msdn.microsoft.com/en-us/library/aa366940(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set in the security descriptor for the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx). By default, this RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set is granted to an administrator.
* “[reset lockout counter after](http://msdn.microsoft.com/en-us/library/cc232773.aspx)” domain policy (LockoutObservationWindow);
  + By default, the value for the “reset lockout counter after” policy is 30 minutes.
  + The Windows OS SAM has the interface of [SamrSetInformationDomain() for the DomainLockoutInformation DomainInformationClass](http://msdn.microsoft.com/en-us/library/cc245788.aspx) for an authorized subject to set the “reset lockout counter after” domain policy of its account domain. This interface requires the caller subject to possess the [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right. By default, this [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right is granted to an administrator.
  + The Windows OS LDAP server has the LDAP interface for an authorized subject to set the “reset lockout counter after” domain policy of the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) in the Active Directory using the [LDAP\_MODIFY\_CMD](http://msdn.microsoft.com/en-us/library/aa366940(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set in the security descriptor for the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx). By default, this RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set is granted to an administrator.
* [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) (if the user account is domain based)
  + It contains associated mappings for X.509 public key certificates or external Kerberos user accounts to the user account for the purpose of public key certificate or smart card based user authentication.
  + The Windows OS LDAP server has the LDAP interface for an authorized subject to set the [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) attribute of the [Windows OS user account Active Directory object](http://msdn.microsoft.com/en-us/library/ms683980(VS.85).aspx) in the Active Directory using the [LDAP\_MODIFY\_CMD](http://msdn.microsoft.com/en-us/library/aa366940(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_WRITE\_PROPERTY right in the security descriptor for the [Windows OS user account Active Directory object](http://msdn.microsoft.com/en-us/library/ms683980(VS.85).aspx). By default, this RIGHT\_DS\_WRITE\_PROPERTY right is granted to an administrator.

Consequently, this Commercial Grade OS Requirement Set “3.1.2.6” requirement is met.

## Addressing 3.1.2.7 “The OS shall provide authorized administrators the ability to specify and configure mandatory password composition”

Application Note: Password composition includes attributes related to password make-up (e.g. mix of uppercase/lowercase, letters, numbers, special characters).

This requirement is addressed by the Windows OS as follows.

The Windows OS supports the enforcement of the following policy.

* “[password complexity](http://support.microsoft.com/kb/279890)” domain policy (DOMAIN\_PASSWORD\_COMPLEX)
  + By default, the password complexity domain policy is enabled on Windows OS domain controllers and is disabled on stand-alone Windows OS servers or workstations. Also, by default, member computers of a Windows OS domain follow the configuration of their Windows OS domain controllers.
  + The Windows OS SAM has the interface of [SamrSetInformationDomain() for the DomainPasswordInformation DomainInformationClass](http://msdn.microsoft.com/en-us/library/cc245788.aspx) for an authorized subject to set the password complexity domain policy of its account domain. This interface requires the caller subject to possess the [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right. By default, this [DOMAIN\_WRITE\_PASSWORD\_PARAMS](http://msdn.microsoft.com/en-us/library/cc211797.aspx) right is granted to an administrator.
  + The Windows OS LDAP server has the LDAP interface for an authorized subject to set the password complexity domain policy of the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx) in the Active Directory using the [LDAP\_MODIFY\_CMD](http://msdn.microsoft.com/en-us/library/aa366940(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set in the security descriptor for the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx). By default, this RIGHT\_DS\_WRITE\_PROPERTY right for the GUID\_PS\_DOMAIN\_PASSWORD property set is granted to an administrator.

Consequently, this Commercial Grade OS Requirement Set “3.1.2.7” requirement is met.

# Meeting the “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Audit Requirements”

In the Commercial Grade OS Requirement Set, there are 5 individual audit requirements under the heading of “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Audit Requirements”. They are listed as “3.1.3.n”, where n = 1, 2, 3, 4, and 5.

## Addressing 3.1.3.1 “The OS shall provide the ability to audit the creation and management of user accounts”

This requirement is addressed by the Windows OS as follows.

The Windows OS SAM is responsible to generate the following security audit records in the case where the user account in question is a locally defined user account object in the Windows OS SAM.

* [Event ID 4720](Event%20ID%204767%20(SE_AUDITID_ACCOUNT_UNLOCKED/SE_AUDITID_ETW_ACCOUNT_UNLOCKED)) “A user account was created” (SE\_AUDITID\_USER\_CREATED/SE\_AUDITID\_ETW\_USER\_CREATED);
* [Event ID 4738](Event%20ID%204767%20(SE_AUDITID_ACCOUNT_UNLOCKED/SE_AUDITID_ETW_ACCOUNT_UNLOCKED)) “A user account was changed” (SE\_AUDITID\_USER\_CHANGE/ SE\_AUDITID\_ETW\_USER\_CHANGE);
* [Event ID 4726](Event%20ID%204767%20(SE_AUDITID_ACCOUNT_UNLOCKED/SE_AUDITID_ETW_ACCOUNT_UNLOCKED)) “A user account was deleted” (SE\_AUDITID\_USER\_DELETED/ SE\_AUDITID\_ETW\_USER\_DELETED);
* [Event ID 4722](Event%20ID%204767%20(SE_AUDITID_ACCOUNT_UNLOCKED/SE_AUDITID_ETW_ACCOUNT_UNLOCKED)) “A user account was enabled” (SE\_AUDITID\_USER\_ENABLED/ SE\_AUDITID\_ETW\_USER\_ENABLED);
* [Event ID 4725](Event%20ID%204767%20(SE_AUDITID_ACCOUNT_UNLOCKED/SE_AUDITID_ETW_ACCOUNT_UNLOCKED)) “A user account was disabled” (SE\_AUDITID\_USER\_DISABLED/ SE\_AUDITID\_ETW\_USER\_DISABLED);
* [Event ID 4781](Event%20ID%204767%20(SE_AUDITID_ACCOUNT_UNLOCKED/SE_AUDITID_ETW_ACCOUNT_UNLOCKED)) “The name of an account was changed” (SE\_AUDITID\_ACCOUNT\_NAME\_CHANGE/SE\_AUDITID\_ETW\_ACCOUNT\_NAME\_CHANGE),

with the following informational items when available:

* Caller[[7]](#footnote-8) User Name:
* Caller Domain:
* Caller User SID:
* (Target) Account Name:
* (Target) Domain:
* (Target) Account SID:
* Sam Account Name:
* Display Name:
* User Principal Name:
* Home Directory:
* Home Drive:
* Script Path:
* Profile Path:
* Password Last Set:
* Account Expires:
* Primary Group ID:
* AllowedToDelegateTo:
* Old UAC Value:
* New UAC Value:
* User Account Control:
* User Parameters:
* Logon Hours:.

In addition, if the user account in question is a domain wide user account object in the Active Directory, then the Active Directory generates the following security audit records.

* [Event ID 5137](http://support.microsoft.com/kb/947226) “A directory service object was created” (SE\_AUDITID\_ETW\_DS\_OBJECT\_CREATE\_value);
* [Event ID 5136](http://support.microsoft.com/kb/947226) “A directory service object was modified” (SE\_AUDITID\_ETW\_DS\_OBJECT\_MODIFY\_value);
* [Event ID 5139](http://support.microsoft.com/kb/947226) “A directory service object was moved” (SE\_AUDITID\_ETW\_DS\_OBJECT\_MOVE\_value);
* [Event ID 5141](http://support.microsoft.com/kb/947226) “A directory service object was deleted” (SE\_AUDITID\_ETW\_DS\_OBJECT\_DELETE\_value);
* [Event ID 5138](http://support.microsoft.com/kb/947226) “A directory service object was undeleted” (SE\_AUDITID\_ETW\_DS\_OBJECT\_UNDELETE\_value),

with the following informational items when available:

* Caller[[8]](#footnote-9) User Name:
* Caller Domain:
* Caller Logon ID:
* Target object distinguished name:
* Target object new distinguished name:
* Target attribute(s) and their old and new values:.

Consequently, this Commercial Grade OS Requirement Set “3.1.3.1” requirement is met.

## Addressing 3.1.3.2 “The OS shall provide the ability to audit the initialization and modification of user security attributes”

This requirement is addressed by the Windows OS as follows.

The Windows OS SAM is responsible to generate the following security audit records for group objects regardless whether a group object in question is a domain wide group object in the Active Directory or a locally defined group object in the Windows OS SAM.

* [Event ID 4731](http://support.microsoft.com/kb/947226) “A security-enabled local group was created” (SE\_AUDITID\_LOCAL\_GROUP\_CREATED/SE\_AUDITID\_ETW\_LOCAL\_GROUP\_CREATED);
* [Event ID 4735](http://support.microsoft.com/kb/947226) “A security-enabled local group was changed” (SE\_AUDITID\_LOCAL\_GROUP\_CHANGE/SE\_AUDITID\_ETW\_LOCAL\_GROUP\_CHANGE);
* [Event ID 4732](http://support.microsoft.com/kb/947226) “A member was added to a security-enabled local group” (SE\_AUDITID\_LOCAL\_GROUP\_ADD/SE\_AUDITID\_ETW\_LOCAL\_GROUP\_ADD), indicating
  + Name of the member added;
  + ID of the member added;
* [Event ID 4733](http://support.microsoft.com/kb/947226) “A member was removed from a security-enabled local group” (SE\_AUDITID\_LOCAL\_GROUP\_REM/SE\_AUDITID\_ETW\_LOCAL\_GROUP\_REM), indicating
  + Name of the member removed;
  + ID of the member removed;
* [Event ID 4734](http://support.microsoft.com/kb/947226) “A security-enabled local group was deleted” (SE\_AUDITID\_LOCAL\_GROUP\_DELETED/SE\_AUDITID\_ETW\_LOCAL\_GROUP\_DELETED);
* [Event ID 4727](http://support.microsoft.com/kb/947226) “A security-enabled global group was created” (SE\_AUDITID\_GLOBAL\_GROUP\_CREATED/SE\_AUDITID\_ETW\_GLOBAL\_GROUP\_CREATED);
* [Event ID 4737](http://support.microsoft.com/kb/947226) “A security-enabled global group was changed” (SE\_AUDITID\_GLOBAL\_GROUP\_CHANGE/SE\_AUDITID\_ETW\_GLOBAL\_GROUP\_CHANGE);
* [Event ID 4728](http://support.microsoft.com/kb/947226) “A member was added to a security-enabled global group” (SE\_AUDITID\_GLOBAL\_GROUP\_ADD/SE\_AUDITID\_ETW\_GLOBAL\_GROUP\_ADD), indicating
  + Name of the member added;
  + ID of the member added;
* [Event ID 4729](http://support.microsoft.com/kb/947226) “A member was removed from a security-enabled global group” (SE\_AUDITID\_GLOBAL\_GROUP\_REM/SE\_AUDITID\_ETW\_GLOBAL\_GROUP\_REM), indicating
  + Name of the member removed;
  + ID of the member removed;
* [Event ID 4730](http://support.microsoft.com/kb/947226) “A security-enabled global group was deleted” (SE\_AUDITID\_GLOBAL\_GROUP\_DELETED/ SE\_AUDITID\_ETW\_GLOBAL\_GROUP\_DELETED);
* [Event ID 4754](http://support.microsoft.com/kb/947226) “A security-enabled universal group was created” (SE\_AUDITID\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_CREATED/ SE\_AUDITID\_ETW\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_CREATED);
* [Event ID 4755](http://support.microsoft.com/kb/947226) “A security-enabled universal group was changed” (SE\_AUDITID\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_CHANGE/ SE\_AUDITID\_ETW\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_CHANGE),
* [Event ID 4756](http://support.microsoft.com/kb/947226) “A member was added to a security-enabled universal group” (SE\_AUDITID\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_ADD/ SE\_AUDITID\_ETW\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_ADD), indicating
  + Name of the member added;
  + ID of the member added;
* [Event ID 4757](http://support.microsoft.com/kb/947226) “A member was removed from a security-enabled universal group” (SE\_AUDITID\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_REM/ SE\_AUDITID\_ETW\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_REM), indicating
  + Name of the member removed;
  + ID of the member removed;
* [Event ID 4758](http://support.microsoft.com/kb/947226) “A security-enabled universal group was deleted” (SE\_AUDITID\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_DELETED/ SE\_AUDITID\_ETW\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_DELETED);

with the following informational items when available:

* Caller[[9]](#footnote-10) User Name:
* Caller Domain:
* Caller User SID:
* (Target) Account Name (for the group):
* (Target) Domain (for the group):
* (Target) Account SID (for the group):.

Furthermore, the Windows OS SAM also generates

* [Event ID 4723](http://support.microsoft.com/kb/947226) “An attempt was made to change an account's password” (SE\_AUDITID\_USER\_PWD\_CHANGED/SE\_AUDITID\_ETW\_USER\_PWD\_CHANGED);
* [Event ID 4724](http://support.microsoft.com/kb/947226) “An attempt was made to reset an account's password” (SE\_AUDITID\_USER\_PWD\_SET/SE\_AUDITID\_ETW\_USER\_PWD\_SET);
* [Event ID 4767](http://support.microsoft.com/kb/947226) “A user account was unlocked” (SE\_AUDITID\_ACCOUNT\_UNLOCKED/SE\_AUDITID\_ETW\_ACCOUNT\_UNLOCKED),

with the following informational items when available:

* Caller[[10]](#footnote-11) User Name:
* Caller Domain:
* Caller User SID:
* (Target) Account Name:
* (Target) Domain:
* (Target) Account SID:.

Finally, the policy database component of the Windows OS authentication service is responsible to generate the following security audit records when the Windows OS privilege (i.e. user right) and system access assignment policy is changed.

* [Event ID 4704](http://support.microsoft.com/kb/947226) “A user right was assigned” (SE\_AUDITID\_ETW\_USER\_RIGHT\_ASSIGNED\_value)
  + Specific rights (i.e. Windows OS privileges) added;
* [Event ID 4705](http://support.microsoft.com/kb/947226) “A user right was removed” (SE\_AUDITID\_ETW\_USER\_RIGHT\_REMOVED\_value)
  + Specific rights (i.e. Windows OS privileges) removed;
* [Event ID 4717](http://support.microsoft.com/kb/947226) “System security access was granted to an account” (SE\_AUDITID\_ETW\_SYSTEM\_ACCESS\_GRANTED\_value)
  + Specific security access added;
* [Event ID 4718](http://support.microsoft.com/kb/947226) “System security access was removed from an account” (SE\_AUDITID\_ETW\_SYSTEM\_ACCESS\_REMOVED\_value)
  + Specific security access removed,

with the following informational items when available:

* Caller[[11]](#footnote-12) User Name:
* Caller Domain:
* Caller User ID:
* Target account ID for privilege or system access addition or removal:.

Consequently, this Commercial Grade OS Requirement Set “3.1.3.2” requirement is met.

## Addressing 3.1.3.3 “The OS shall provide the ability to audit the initialization and modification of authentication mechanism attributes”

This requirement is addressed by the Windows OS as follows.

The Windows OS SAM is responsible to generate the following security audit record for the modification of domain policies described in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.2.6” requirement.

* [Event ID 4739](http://support.microsoft.com/kb/947226) “Domain Policy was changed” (SE\_AUDITID\_DOMAIN\_POLICY\_CHANGE/SE\_AUDITID\_ETW\_DOMAIN\_POLICY\_CHANGE), with the following informational items when available:
  + Caller[[12]](#footnote-13) User Name:
  + Caller Domain:
  + Caller User SID:
  + (Target) Domain Name:
  + (Target) Domain SID:
  + Min. Password Age:
  + Max. Password Age:
  + Lockout Threshold:
  + Lockout Observation Window:
  + Lockout Duration:
  + Min. Password Length:
  + Password History Length:
  + Password Properties (which indicates the existence of the DOMAIN\_PASSWORD\_COMPLEX flags for the [password complexity](http://support.microsoft.com/kb/279890) policy):.

In addition, if the user account in question is a domain wide user account object in the Active Directory, then the Active Directory generates the following security audit record for the case where the [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) attribute of the user account object is changed.

* [Event ID 5136](http://support.microsoft.com/kb/947226) “A directory service object was modified” (SE\_AUDITID\_ETW\_DS\_OBJECT\_MODIFY\_value), with the following informational items when available:
  + Caller[[13]](#footnote-14) User Name:
  + Caller Domain:
  + Caller User SID:
  + Directory Server Name:
  + Directory Object Name:
  + Directory Object GUID:
  + Directory Object Class:
  + Directory Object Attribute Name:
  + Directory Object Attribute Value:.

Consequently, this Commercial Grade OS Requirement Set “3.1.3.3” requirement is met.

## Addressing 3.1.3.4 “The OS shall provide the ability to audit when a user’s consecutive unsuccessful authentication attempts meets or exceeds the administrator-configured positive integer”

This requirement is addressed by the Windows OS as follows.

As described in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.5” requirement, the Windows OS SAM is responsible to generate the [[Event ID 4740](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ACCOUNT\_AUTO\_LOCKED/SE\_AUDITID\_ETW\_ACCOUNT\_AUTO\_LOCKED)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record to record the event that the specific user account is locked out in the case where the user account was not previously locked.

* [Event ID 4740](http://support.microsoft.com/kb/947226) “A user account was locked out” (SE\_AUDITID\_ACCOUNT\_AUTO\_LOCKED/SE\_AUDITID\_ETW\_ACCOUNT\_AUTO\_LOCKED), with the following informational items when available:
  + (Target) Account Name:
  + (Target) Account Domain:
  + (Target) User SID:.

Consequently, this Commercial Grade OS Requirement Set “3.1.3.4” requirement is met.

## Addressing 3.1.3.5 “The OS shall provide the ability to audit all user attempts to identify and authenticate to the system”

This requirement is addressed by the Windows OS as follows.

### Logon audit records generated at a Windows OS domain controller

The Kerberos KDC residing on a Windows OS domain controller is responsible to generate the following security audit records for the specific events when handling the corresponding [Kerberos protocol service request messages](http://msdn.microsoft.com/en-us/library/aa378048(VS.85).aspx) in the user authentication processing of the logging on of a domain wide user account.

* [Event ID 4770](http://support.microsoft.com/kb/947226) “Kerberos pre-authentication failed” (SE\_AUDITID\_PREAUTH\_FAILURE/SE\_AUDITID\_ETW\_PREAUTH\_FAILURE), with the following informational items when available:
  + Target User Name:
  + Target User SID:
  + Ticket Options:
  + Status:
  + PreAuthType:
  + Ticket Encryption Type:
  + IP Address:
  + IP Port:
  + Certificate Issuer Name:
  + Certificate Serial Number:
* [Event ID 4768](http://support.microsoft.com/kb/947226) “A Kerberos authentication ticket (TGT) was requested” (SE\_AUDITID\_AS\_TICKET/SE\_AUDITID\_ETW\_AS\_TICKET) for success and failure, with the following informational items when available:
  + Target User Name:
  + Target Domain Name:
  + Target User SID:
  + Ticket Options:
  + Status:
  + PreAuthType:
  + Ticket Encryption Type:
  + IP Address:
  + IP Port:
  + Certificate Issuer Name:
  + Certificate Serial Number:
* [Event ID 4769](http://support.microsoft.com/kb/947226) “A Kerberos service ticket was requested” (SE\_AUDITID\_TGS\_TICKET\_REQUEST/SE\_AUDITID\_ETW\_TGS\_TICKET\_REQUEST) for success and failure, with the following informational items when available:
  + Target User Name:
  + Target Domain Name:
  + Ticket Options:
  + Ticket Encryption Type:
  + IP Address:
  + IP Port:
  + Status:
  + Logon GUID:
* [Event ID 4770](http://support.microsoft.com/kb/947226) “A Kerberos service ticket was renewed” (SE\_AUDITID\_TICKET\_RENEW\_SUCCESS/SE\_AUDITID\_ETW\_TICKET\_RENEW\_SUCCESS), with the following informational items when available:
  + Target User Name:
  + Target Domain Name:
  + Ticket Options:
  + Ticket Encryption Type:
  + IP Address:
  + IP Port:
* [Event ID 4774](http://support.microsoft.com/kb/947226) “An account was mapped for logon” (SE\_AUDITID\_ACCOUNT\_MAPPED/SE\_AUDITID\_ETW\_ACCOUNT\_MAPPED) for success when the value of the [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) attribute is used to achieved to the mapping, with the following informational items when available:
  + Mapping By:
  + Client User Name:
  + Mapped Name:

The [Schannel security provider](http://msdn.microsoft.com/en-us/library/aa380123(VS.85).aspx) residing in a Windows OS domain controller is responsible to generate the following security audit records for the specific events when handling the corresponding [RFC 2246 Transport Layer Security (TLS)](http://msdn2.microsoft.com/en-us/library/aa380123(VS.85).aspx) protocol messages in the user authentication processing of the logging on of a domain wide user account.

* [Event ID 4774](http://support.microsoft.com/kb/947226) “An account was mapped for logon” (SE\_AUDITID\_ACCOUNT\_MAPPED/SE\_AUDITID\_ETW\_ACCOUNT\_MAPPED) for success and failure, with the following informational items when available:
  + Mapping By:
  + Client User Name:
  + Mapped Name:

The [Digest security provider](http://msdn.microsoft.com/en-us/library/aa378745(VS.85).aspx) residing in a Windows OS domain controller is responsible to generate the following security audit records for the specific events when handling the corresponding [RFC 2617 Digest Access Protocol](http://msdn2.microsoft.com/en-us/library/aa378745(VS.85).aspx) messages in the user authentication processing of the logging on of a domain wide user account.

* [Event ID 4776](http://support.microsoft.com/kb/947226) “The domain controller attempted to validate the credentials for an account” (SE\_AUDITID\_ACCOUNT\_LOGON/SE\_AUDITID\_ETW\_ACCOUNT\_LOGON) for success and failure, with the following informational items when available:
  + Security Package/Provider Name:
  + Target User Name:
  + Status:

### Logon audit records generated at an individual Windows OS machine

A security provider residing in the Windows OS Authentication Service (aka lsass.exe) is responsible to generate the following security audit records for the specific events when handling the user authentication processing of the logging on of a user account occurring in the local Windows OS machine.

* [Event ID 4625](http://support.microsoft.com/kb/947226) “An account failed to log on” (SE\_AUDITID\_ETW\_LOGON\_FAILURE\_value), with the following informational items when available:
  + Target User Name:
  + Target Domain Name:
  + Target User SID:
  + Status:
  + Failure Reason (being one of the following):
    - “An Error occurred during Logon” due to [STATUS\_BAD\_VALIDATION\_CLASS](http://msdn.microsoft.com/en-us/library/cc704588.aspx);
    - “The specified user account has expired” due to [STATUS\_ACCOUNT\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx);
    - “The NetLogon component is not active” due to [STATUS\_NETLOGON\_NOT\_STARTED](http://msdn.microsoft.com/en-us/library/cc704588.aspx);
    - “Account locked out” due to [STATUS\_ACCOUNT\_LOCKED\_OUT](http://msdn.microsoft.com/en-us/library/cc704588.aspx);
    - “The user has not been granted the requested logon type at this machine” due to [STATUS\_LOGON\_TYPE\_NOT\_GRANTED](http://msdn.microsoft.com/en-us/library/cc704588.aspx);
    - “The specified account's password has expired” due to [STATUS\_PASSWORD\_MUST\_CHANGE](http://msdn.microsoft.com/en-us/library/cc704588.aspx);
    - “The specified account's password has expired” due to [STATUS\_PASSWORD\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) and [STATUS\_ACCOUNT\_RESTRICTION](http://msdn.microsoft.com/en-us/library/cc704588.aspx);
    - “Account currently disabled” due to [STATUS\_ACCOUNT\_DISABLED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) and [STATUS\_ACCOUNT\_RESTRICTION](http://msdn.microsoft.com/en-us/library/cc704588.aspx);
    - “Account logon time restriction violation” due to [STATUS\_INVALID\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/cc704588.aspx) and [STATUS\_ACCOUNT\_RESTRICTION](http://msdn.microsoft.com/en-us/library/cc704588.aspx);
    - “Unknown user name or bad password” due to any other random status pairing with [STATUS\_ACCOUNT\_RESTRICTION](http://msdn.microsoft.com/en-us/library/cc704588.aspx);
    - “Unknown user name or bad password” due to [STATUS\_LOGON\_FAILURE](http://msdn.microsoft.com/en-us/library/cc704588.aspx) and [STATUS\_WRONG\_PASSWORD](http://msdn.microsoft.com/en-us/library/cc704588.aspx)/[STATUS\_NO\_SUCH\_USER](http://msdn.microsoft.com/en-us/library/cc704588.aspx);
    - “Domain sid inconsistent” due to [STATUS\_LOGON\_FAILURE](http://msdn.microsoft.com/en-us/library/cc704588.aspx) and [STATUS\_DOMAIN\_TRUST\_INCONSISTENT](http://msdn.microsoft.com/en-us/library/cc704588.aspx);
    - “An Error occurred during Logon” due to any other random status,
  + SubStatus:
  + LogonType:
  + Logon Process Name :
  + Security Package/Provider Name:
  + IP Address:
  + IP Port:
* [Event ID 4624](http://support.microsoft.com/kb/947226) “An account was successfully logged on” (SE\_AUDITID\_ETW\_SUCCESSFUL\_LOGON\_value
* ), with the following informational items when available:
  + Target User Name:
  + Target Domain Name:
  + Target User SID:
  + Status:
  + LogonType:
  + Logon Process Name :
  + Security Package/Provider Name:
  + Logon GUID:
  + IP Address:
  + IP Port:.

### Specific group assignment to a logged on user audit record

The Windows OS supports the situation when an administrator wishes to know if a member of a specific group logs on successful to a Windows OS machine. An administrator edits the below registry key value by including a group SID of the specific group that s/he is interested in for the event of a logging on from a group member.

* HKEY\_LOCAL\_MACHINE\ System\CurrentControlSet\Control\Lsa\Audit\SpecialGroups.

After the Windows OS Authentication Service (aka lsass.exe) has determined the successful logging on to a user account, the Windows OS Authentication Service is aware of the expanded list of groups where the logged on user account belongs to directly or indirectly. If the user account’s expanded list of groups includes a specific group belonging to the above “SpecialGroups” registry key value, then the Windows OS Authentication Service generates an [Event ID 4964](http://support.microsoft.com/kb/947226) “Special groups have been assigned to a new logon” (SE\_AUDITID\_ETW\_SPECIAL\_GROUP\_LOGON\_value) security audit record, with the following informational items when available:

* + Machine Name:
  + Machine Domain Name:
  + Machine SID:
  + Machine Logon ID:
  + Machine Logon GUID:
  + Target User Name:
  + Target User Domain Name:
  + Target User SID:
  + Target User Logon ID:
  + Target User Logon GUID:.

By default, the above “SpecialGroups” registry key value is empty, and only an administrator is allowed to modify the associated value data. When this registry key value is changed, the Windows OS Authentication Service is notified so that a reload is necessary. After it reloads the value data associated with the “SpecialGroups” registry key value, the Windows OS Authentication Service generates an [Event ID 4908](http://support.microsoft.com/kb/947226) “Special Groups Logon table modified” (SE\_AUDITID\_ETW\_SPECIAL\_GROUPS\_LOGON\_TABLE\_CREATION\_value) security audit record with an indication of the current list of group SIDs in the registry key value.

Consequently, this Commercial Grade OS Requirement Set “3.1.3.5” requirement is met.

# Meeting the “Identification and Authentication User Interface Security Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 8 individual functional requirements under the heading of “Identification and Authentication User Interface Security Functional Requirements”. They are listed as “3.2.1.n”, where n = 1, 2, 3, 4, 5, 6, 7, and 8.

## Addressing 3.2.1.1 “Before establishing an interactive session, the OS shall display an authorized administrator specified advisory notice”

This requirement is addressed by the Windows OS as follows.

As described in the [Microsoft publication: “Security Functional Assertions of the “User Interaction based on Windowing” Scenario of a Modern Operating System”](http://download.microsoft.com/download/e/d/b/edbb17fb-580d-49a4-b66c-8726cf446a86/User%20Interaction.doc), every instance of the window logon state maintaining service (aka winlogon.exe) contains, among other states:

* the “Welcome” state;
* the “Display legal notices” state;
* the “Request logon credentials” state;
* the “End” state, which represents the winlogon.exe instance’s termination.



### Transitions from the “Welcome” state

The “Welcome” state supports the following state transition, among the others.

* If the “Welcome” state transitions to the “Display legal notices” state, then there must be either:
  + the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
  + the notification of a trusted path simulation from an authorized subject.

### “Display legal notices” state

Upon entering the “Display legal notices” state, the window logon state maintaining service requests the window logon user interface service (aka LogonUI.exe) to display the “legal notices” message box message on the secure display area to collect the interactive user’s response.

### Transitions from the “Display legal notices” state

The “Display legal notices” state supports the following state transitions, among the others.

* If the “Display legal notices” state transitions to the “Request logon credentials” state, then either:
  + there must have been the interactive user’s acknowledgement as his response to the displayed “legal notices” message box message on the secure display area;
  + the administrator must have not specified any “legal notices”.
* If the “Display legal notices” state transitions to the “Welcome” state, then either:
  + the Local Window Terminal Service must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in;
  + the local shutdown initiation service (as part of ininit.exe) must have notified a logoff or shutdown;
  + a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + there must have been a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service (aka LogonUI.exe) to display the below messages:
    - “Windows is shutting down...” corresponding to data saving;
    - “Undocking and going to sleep...”;
    - “Hibernating...”
    - “Going to sleep...” corresponding to the entering of the “standing by” Windows OS system power state;
  + in the local terminal session case, there must have been a timeout for receiving the user response through a window dialog displayed by the window logon user interface service;
  + in the local terminal session case, there must have been a failure reported from the window logon user interface service during the display of the “legal notices” message box message.
* If the “Display legal notices” state transitions to the “End” state, then either:
  + in the remote terminal session case, there must have been a timeout for receiving the user response through a window dialog displayed by the window logon user interface service;
  + in the remote terminal session case, there must have been a failure reported from the window logon user interface service during the display of the “legal notices” message box message.

Because the entering of the “Request logon credentials” state is critical for establishing an interactive (logged on) session, this Commercial Grade OS Requirement Set “3.2.1.1” requirement is met.

## Addressing 3.2.1.2 “The OS shall allow user-initiated locking of an interactive session by performing a specific set of actions”

The Commercial Grade OS Requirement Set requires the following specific set of actions being performed by the OS:

1. Clearing or overwriting display devices, making the current contents unreadable;
2. Disabling any activity from the user’s data access/display devices other than unlocking the interactive session.

This requirement is addressed by the Windows OS as follows.

As described in the [Microsoft publication: “Security Functional Assertions of the “User Interaction based on Windowing” Scenario of a Modern Operating System”](http://download.microsoft.com/download/e/d/b/edbb17fb-580d-49a4-b66c-8726cf446a86/User%20Interaction.doc), every instance of the window logon state maintaining service (aka winlogon.exe) contains, among other states:

* the “User logged on” state;
* the “Trusted path activated” state;
* the “Initiate lock on display areas” state;
* the “Locked” state;
* the “Request to unlock” state;
* the “Inactivity timeout handler during locked” state;
* the “Session disconnected during locked” state;
* the “Hibernate during locked” state;
* the “Attempt to unlock” state;
* the “Report unlock success” state;
* the “Report unlock failure” state;
* the “Report last logon to user after unlock” state.

### Transitions from the “User logged on” state

The “User logged on” state supports the following state transition, among the others.

* If the “User logged on” state transitions to the “Trusted path activated” state, then there must be either:
  + the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
  + the notification of a trusted path simulation from an authorized subject.



### “Trusted path activated” state

The instance of the window logon state maintaining service conducts the following:

* switches to the secure display area;
* requests the window logon user interface service to display the following option, among the others, in a dialog on the secure display area for the interactive user to select:
  + lock the display areas from visibility;

upon entering the “Trusted path activated” state, in the case where there is no pending request for any of the following:

* credential backup;
* credential restore;
* prompting for credentials.

### Transitions from the “Trusted path activated” state

The “Trusted path activated” state supports the following state transition, among the others.

* If the “Trusted path activated” state transitions to the “Initiate lock on display areas” state, then either:
  + the user must have selected the “lock the display areas from visibility” option;
  + there must have been a notification of an authorized subject’s initiation of a locking of the display areas or there must have been a request originated from an authorized subject to lock the display areas.

### “Initiate lock on display areas” state

The instance of the window logon state maintaining service conducts the following:

* switches to the user interactive application display area,

upon entering the “Initiate lock on display areas” state, in the case where the administrator specified “[display area locking](http://technet2.microsoft.com/windowsserver/en/library/c7410c7f-d6cf-400b-9604-f1e2b4fe1d2c1033.mspx?mfr=true)” policy is disabled.

The instance of the window logon state maintaining service conducts the following:

* switches to the secure display area;
* generates the [Event ID 4800](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_WORKSTATION\_LOCKED) “The (workstation) display areas are locked” audit record in the hard audit store;

upon entering the “Initiate lock on display areas” state, in the case where the administrator specified “[display area locking](http://technet2.microsoft.com/windowsserver/en/library/c7410c7f-d6cf-400b-9604-f1e2b4fe1d2c1033.mspx?mfr=true)” policy is enabled.

The “[display area locking](http://technet2.microsoft.com/windowsserver/en/library/c7410c7f-d6cf-400b-9604-f1e2b4fe1d2c1033.mspx?mfr=true)” policy is enabled by default. This policy is defined by the following registry key value.

* HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\DisableLockWorkstation.

### Transitions from the “Initiate lock on display areas” state

The “Initiate lock on display areas” state supports the following state transitions.

* If the “Initiate lock on display areas” state transitions to the “User logged on” state, then
  + the instance of the window logon state maintaining service must have switched to the user interactive application display area.
* If the “Initiate lock on display areas” state transitions to the “Locked” state, then
  + the instance of the window logon state maintaining service must have switched to the secure display area.

### “Locked” state

Due to the switching to the secure display area before entering the “Locked” state, the user interactive application display area, belonging to the instance of the window logon state maintaining service, is not visible and is not receiving user input.

In the local window terminal session case, the instance of the window logon state maintaining service requests the window logon user interface service to display one of following “Locked” messages and options on the secure display area with the specified user identity of the logged on user account of the window terminal session that is being locked, upon entering the “Locked” state, in the case where the administrator specified “[disable trusted path](http://technet2.microsoft.com/windowsserver/en/library/83e7e44d-84fe-4e8b-9e5c-df97609eb8601033.mspx?mfr=true)” policy is disabled.

* A message: “Press CTRL + ALT + DELETE to unlock this computer. <user name> is logged on”, in the case of a non tablet machine;
* A message: “Press CTRL + ALT + DELETE or use the Windows Security button to unlock this Tablet PC. <user name> is logged on”, in the case of a tablet machine;
* An option to switch to another local window terminal session to allow a different user to log on or switch to another logon user account’s local window terminal session, in the case where the session switch behavior is allowed by the administrator specified “[Hide entry points for Fast User Switching](http://support.microsoft.com/kb/329885)” policy;
* An option to launch the accessibility tool.

By default, the administrator specified “[disable trusted path](http://technet2.microsoft.com/windowsserver/en/library/83e7e44d-84fe-4e8b-9e5c-df97609eb8601033.mspx?mfr=true)” policy is disabled.

By default, the administrator specified “[Hide entry points for Fast User Switching](http://support.microsoft.com/kb/329885)” policy is disabled.



### Transitions from the “Locked” state

The “Locked” state supports the following state transitions.

* If the “Locked” state transitions to the “Request to unlock” state, then either:
  + the instance of the window logon state maintaining service must operate in the remote window terminal session case;
  + the “[disable trusted path](http://technet2.microsoft.com/windowsserver/en/library/83e7e44d-84fe-4e8b-9e5c-df97609eb8601033.mspx?mfr=true)” policy must have been enabled;
  + there must have been either:
    - the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
    - the notification of a trusted path simulation from an authorized subject.
* If the “Locked” state transitions to the “Inactivity timeout handler during locked” state, then
  + there must have been a notification of the user input inactivity timeout which has been mediated by the local Windows OS window manager of the same window terminal session.
* If the “Locked” state transitions to the “Session disconnected during locked” state, then
  + the Local Window Terminal Service (LWTS) (lsm.exe) must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in.
* If the “Locked” state transitions to the “Hibernate during locked” state, then
  + there must have been a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service to display the below messages:
    - “Windows is shutting down...”;
    - “Undocking and going to sleep...”;
    - “Hibernating...”;
    - “Going to sleep...”, corresponding to the entering of the “standing by” Windows OS system power state.
* If the “Locked” state transitions to the “Welcome” state, then
  + either:
    - the local Windows OS shutdown initiation service (as part of wininit.exe) must have notified a logoff or shutdown;
    - a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + the instance of the window logon state maintaining service must have switched to the secure display area;
  + the instance of the window logon state maintaining service must have terminated any remaining applications (if any) which belong to the authenticated interactive user account;
  + the instance of the window logon state maintaining service generates the [Event ID 4647](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_BEGIN\_LOGOFF.id) “User initiated logoff” audit record in the hard audit store;
  + the instance of the window logon state maintaining service must have requested its local Windows OS window manager to assign no one (i.e. the null [locally unique ID (LUID)](http://msdn.microsoft.com/en-us/library/aa379261.aspx)) as the interactive logged on user of the window manager’s corresponding window terminal session;
  + the instance of the window logon state maintaining service must have revoked any of the following rights to the display areas (residing in the window terminal session where the service resides in) that may have been granted to the previously logged on user account:
    - the specific rights that are necessary in some of the Windows OS window manager operations:
      * the write attributes access right ([DESKTOP\_WRITEOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx)) to all the display areas;
      * the read attributes access right ([DESKTOP\_READOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx) ) to all the display areas;
      * the display area creation access right ([WINSTA\_CREATEDESKTOP](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the display area enumeration access right ([WINSTA\_ENUMDESKTOPS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the clipboard access right to all the display areas ([WINSTA\_ACCESSCLIPBOARD](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS graphic device interface (GDI) Manager (win32k.sys) operations:
      * the screen content access right to all the display areas ([WINSTA\_READSCREEN](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS kernel executive manager (ex.lib) operations (e.g. registering a character string to the global atom table maintained by the Windows OS kernel and receiving a unique value [called an atom for identifying the registered string]):
      * the global atom manipulation access right ([WINSTA\_ACCESSGLOBALATOMS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx)) to all the display areas;
  + the instance of the window logon state maintaining service must have reset the security of the user interactive application display area so that no subject of any user account is granted the permission for accessing the user interactive application display area.

### “Request to unlock” state

The instance of the window logon state maintaining service requests the window logon user interface service to display the “display area unlock” frame on the secure display area for an interactive user to attempt to unlock the locked window terminal session, using the user’s credential information (such as the user name with a password or the smart card PIN) corresponding to a credential provider installed on the local machine, upon entering the “Request to unlock” state.

### Transitions from the “Request to unlock” state

The “Request to unlock” state supports the following state transitions.

* If the “Request to unlock” state transitions to the “Attempt to unlock” state, then
  + the interactive user must have supplied the user’s credential information.
* If the “Request to unlock” state transitions to the “Locked” state, then either:
  + there must have been a failure (including the user’s cancellation) reported from the window logon user interface service during the display of the “display area unlock” frame;
  + there must have been a timeout for receiving the user response through a window dialog displayed by window logon user interface service.
* If the “Request to unlock” state transitions to the “Locked” state and then to the “Inactivity timeout handler during locked” state[[14]](#footnote-15), then
  + there must have been a notification of the user input inactivity timeout which has been mediated by the local Windows OS window manager of the same window terminal session.
* If the “Request to unlock” state transitions to the “Locked” state and then to the “Session disconnected during locked” state[[15]](#footnote-16), then
  + the Local Window Terminal Service (LWTS) (lsm.exe) must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in.
* If the “Request to unlock” state transitions to the “Locked” state and then to the “Hibernate during locked” state[[16]](#footnote-17), then
  + there must have been a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service to display the below messages:
    - “Windows is shutting down...”;
    - “Undocking and going to sleep...”;
    - “Hibernating...”;
    - “Going to sleep...”, corresponding to the entering of the “standing by” Windows OS system power state.
* If the “Request to unlock” state transitions to the “Welcome” state, then
  + either:
    - the local Windows OS shutdown initiation service (as part of wininit.exe) must have notified a logoff or shutdown;
    - a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + the instance of the window logon state maintaining service must have switched to the secure display area;
  + the instance of the window logon state maintaining service must have terminated any remaining applications (if any) which belong to the authenticated interactive user account;
  + the instance of the window logon state maintaining service generates the [Event ID 4647](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_BEGIN\_LOGOFF.id) “User initiated logoff” audit record in the hard audit store;
  + the instance of the window logon state maintaining service must have requested its local Windows OS window manager to assign no one (i.e. the null [LUID](http://msdn.microsoft.com/en-us/library/aa379261.aspx)) as the interactive logged on user of the window manager’s corresponding window terminal session;
  + the instance of the window logon state maintaining service must have revoked any of the following rights to the display areas (residing in the window terminal session where the service resides in) that may have been granted to the previously logged on user account:
    - the specific rights that are necessary in some of the Windows OS window manager operations:
      * the write attributes access right ([DESKTOP\_WRITEOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx)) to all the display areas;
      * the read attributes access right ([DESKTOP\_READOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx) ) to all the display areas;
      * the display area creation access right ([WINSTA\_CREATEDESKTOP](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the display area enumeration access right ([WINSTA\_ENUMDESKTOPS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the clipboard access right to all the display areas ([WINSTA\_ACCESSCLIPBOARD](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS graphic device interface (GDI) Manager (win32k.sys) operations:
      * the screen content access right to all the display areas ([WINSTA\_READSCREEN](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS kernel executive manager (ex.lib) operations (e.g. registering a character string to the global atom table maintained by the Windows OS kernel and receiving a unique value [called an atom for identifying the registered string]):
      * the global atom manipulation access right ([WINSTA\_ACCESSGLOBALATOMS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx)) to all the display areas;
  + the instance of the window logon state maintaining service must have reset the security of the user interactive application display area so that no subject of any user account is granted the permission for accessing the user interactive application display area.

### “User input inactivity timeout” policy

The Windows OS supports the “user input inactivity timeout” policy, which consists of an attribute as follows:

* one of the following actions to be taken upon an inactivity timeout:
  + Nothing;
  + Start the screen saver, and the screen saver runs on the screen saver display area or on the user interactive application display area depending on the corresponding “screen saver” policy;
  + Locking of the display areas;
  + Log off;
  + Start a password protected screen saver on the screen saver display area,

through the following registry key value:

* + HKEY\_LOCAL\_MACHINE\Software\Policies\Microsoft\Windows\Control Panel\Desktop\ScreenSaveActive, where
    - the default action is “Nothing”.

By default, only an administrator is allowed to configure the “user input inactivity timeout” policy.

### “Screen saver” policy

The Windows OS supports the “screen saver” policy, which consists of three attributes as follows:

* the name of the screen saver application that is specified by the administrator, through the following registry key value:
  + HKEY\_LOCAL\_MACHINE\Software\Policies\Microsoft\Windows\Control Panel\Desktop\SCRNSAVE.EXE, where
    - the default value is “scrnsave.scr”;
* whether the administrator specified screen saver application process runs on its own screen saver display area or on the user interactive application display area, through the following registry key value:
  + HKEY\_LOCAL\_MACHINE\Software\Policies\Microsoft\Windows\Control Panel\Desktop\ScreenSaverIsSecure, where
    - the default value is “false”;
* the administrator specified length of the screen saver grace period which specifies how much user idle time must elapse before the actual enforcement of the “user input inactivity timeout” policy:
  + HKEY\_LOCAL\_MACHINE\Software\Policies\Microsoft\Windows\Control Panel\Desktop\ScreenSaverGracePeriod, where
    - the default value is 5 seconds.

By default, only an administrator is allowed to configure the “screen saver” policy.

### “Inactivity timeout handler during locked” state

The instance of the window logon state maintaining service conducts the following:

* switches to the screen saver display area;
* starts the administrator specified screen saver application (according to the “screen saver” policy) running in the security context of the authenticated interactive user on the screen saver display area;
* generates the [Event ID 4802](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_SCREENSAVER\_INVOKED) “The screen saver was invoked” audit record in the hard audit store,

upon entering the “Inactivity timeout handler during locked” state, in the case where either

1. the “user input inactivity timeout” policy specifies “Start a password protected screen saver”;
2. the “user input inactivity timeout” policy specifies “Start the screen saver”.

While still remaining in the “Inactivity timeout handler during locked” state, the instance of the window logon state maintaining service waits for the termination of the administrator specified screen saver application (according to the “screen saver” policy) started by the service.

The administrator specified screen saver application process started by the instance of the window logon state maintaining service terminates itself upon receiving the following window messages from the local Windows OS window manager:

* a mouse button is physically pressed ([WM\_LBUTTONDOWN](http://msdn.microsoft.com/en-us/library/ms645607.aspx), [WM\_MBUTTONDOWN](http://msdn.microsoft.com/en-us/library/ms645610(VS.85).aspx), [WM\_RBUTTONDOWN](http://msdn.microsoft.com/en-us/library/ms646242(VS.85).aspx));
* the mouse is physically moved by a specific threshold distance ([WM\_MOUSEMOVE](http://msdn.microsoft.com/en-us/library/ms645616(VS.85).aspx));
* a non-system keyboard key is pressed, where a non-system key is a keyboard key that is pressed when the ALT key is not pressed ([WM\_KEYDOWN](http://msdn.microsoft.com/en-us/library/ms646280(VS.85).aspx));
* the translated character of the non-system key being pressed ([WM\_CHAR](http://msdn.microsoft.com/en-us/library/ms646276(VS.85).aspx)).

The instance of the window logon state maintaining service conducts the following:

* generates the [Event ID 4803](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_SCREENSAVER\_DISMISSED) “The screen saver was dismissed” audit record in the hard audit store;
* switches back to the secure display area,

before exiting from the “Inactivity timeout handler during locked” state, in the case where either

1. the “user input inactivity timeout” policy specifies “Start a password protected screen saver”;
2. the “user input inactivity timeout” policy specifies “Start the screen saver”.

The instance of the window logon state maintaining service conducts the following:

* initiates a forced logoff for the already authenticated interactive user,

before exiting from the “Inactivity timeout handler during locked” state, in the case where either

1. the “user input inactivity timeout” policy specifies “Log off”.

### Transitions from the “Inactivity timeout handler during locked” state

The “Inactivity timeout handler during locked” state supports the following state transitions.

* If the “Inactivity timeout handler during locked” state transitions to the “Locked” state, then
  + the instance of the window logon state maintaining service must have switched back to the secure display area.
* If the “Inactivity timeout handler during locked” state transitions to the “Locked” state and then to the “Session disconnected during locked” state[[17]](#footnote-18), then
  + the Local Window Terminal Service (LWTS) (lsm.exe) must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in.
* If the “Inactivity timeout handler during locked” state transitions to the “Locked” state and then to the “Hibernate during locked” state[[18]](#footnote-19), then
  + there must have been a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service to display the below messages:
    - “Windows is shutting down...”;
    - “Undocking and going to sleep...”;
    - “Hibernating...”;
    - “Going to sleep...”, corresponding to the entering of the “standing by” Windows OS system power state.
* If the “Inactivity timeout handler during locked” state transitions to the “Locked” state and then to the “Request to unlock” state[[19]](#footnote-20), then either:
  + the instance of the window logon state maintaining service must operate in the remote window terminal session case;
  + the “[disable trusted path](http://technet2.microsoft.com/windowsserver/en/library/83e7e44d-84fe-4e8b-9e5c-df97609eb8601033.mspx?mfr=true)” policy must have been enabled;
  + there must have been either:
    - the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
    - the notification of a trusted path simulation from an authorized subject.
* If the “Inactivity timeout handler during locked” state transitions to the “Locked” state and then to the “Welcome” state[[20]](#footnote-21), then
  + either:
    - the local Windows OS shutdown initiation service (as part of wininit.exe) must have notified a logoff or shutdown;
    - a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + the instance of the window logon state maintaining service must have switched to the secure display area;
  + the instance of the window logon state maintaining service must have terminated any remaining applications (if any) which belong to the authenticated interactive user account;
  + the instance of the window logon state maintaining service generates the [Event ID 4647](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_BEGIN\_LOGOFF.id) “User initiated logoff” audit record in the hard audit store;
  + the instance of the window logon state maintaining service must have requested its local Windows OS window manager to assign no one (i.e. the null [LUID](http://msdn.microsoft.com/en-us/library/aa379261.aspx)) as the interactive logged on user of the window manager’s corresponding window terminal session;
  + the instance of the window logon state maintaining service must have revoked any of the following rights to the display areas (residing in the window terminal session where the service resides in) that may have been granted to the previously logged on user account:
    - the specific rights that are necessary in some of the Windows OS window manager operations:
      * the write attributes access right ([DESKTOP\_WRITEOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx)) to all the display areas;
      * the read attributes access right ([DESKTOP\_READOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx) ) to all the display areas;
      * the display area creation access right ([WINSTA\_CREATEDESKTOP](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the display area enumeration access right ([WINSTA\_ENUMDESKTOPS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the clipboard access right to all the display areas ([WINSTA\_ACCESSCLIPBOARD](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS graphic device interface (GDI) Manager (win32k.sys) operations:
      * the screen content access right to all the display areas ([WINSTA\_READSCREEN](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS kernel executive manager (ex.lib) operations (e.g. registering a character string to the global atom table maintained by the Windows OS kernel and receiving a unique value [called an atom for identifying the registered string]):
      * the global atom manipulation access right ([WINSTA\_ACCESSGLOBALATOMS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx)) to all the display areas;
  + the instance of the window logon state maintaining service must have reset the security of the user interactive application display area so that no subject of any user account is granted the permission for accessing the user interactive application display area.

### “Session disconnected during locked” state

The instance of the window logon state maintaining service reacts to one of the following notifications, upon entering the “Session disconnected during locked” state.

* The local Windows OS shutdown initiation service (as part of wininit.exe) notifies a logoff or shutdown.
* A logoff or shutdown is triggered internally within instance of the window logon state maintaining service.
* A notification from the Local Window Terminal Service (LWTS) (lsm.exe) that the window terminal session, where the instance of the window logon state maintaining service resides in, is reconnected by the Local Window Terminal Service.

### Transitions from the “Session disconnected during locked” state

The “Session disconnected during locked” state supports the following state transitions.

* If the “Session disconnected during locked” state transitions to the “User logged on” state, then
  + there must have been a notification from the Local Window Terminal Service (LWTS) (lsm.exe) that the window terminal session, where the instance of the window logon state maintaining service resides in, is reconnected by the Local Window Terminal Service;
  + the instance of the window logon state maintaining service must have switched back to the user interactive application display area.
* If the “Session disconnected during locked” state transitions to the “Welcome” state, then
  + either:
    - the local Windows OS shutdown initiation service (as part of wininit.exe) must have notified a logoff or shutdown;
    - a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + the instance of the window logon state maintaining service must have switched to the secure display area;
  + the instance of the window logon state maintaining service must have terminated any remaining applications (if any) which belong to the authenticated interactive user account;
  + the instance of the window logon state maintaining service generates the [Event ID 4647](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_BEGIN\_LOGOFF.id) “User initiated logoff” audit record in the hard audit store;
  + the instance of the window logon state maintaining service must have requested its local Windows OS window manager to assign no one (i.e. the null [LUID](http://msdn.microsoft.com/en-us/library/aa379261.aspx)) as the interactive logged on user of the window manager’s corresponding window terminal session;
  + the instance of the window logon state maintaining service must have revoked any of the following rights to the display areas (residing in the window terminal session where the service resides in) that may have been granted to the previously logged on user account:
    - the specific rights that are necessary in some of the Windows OS window manager operations:
      * the write attributes access right ([DESKTOP\_WRITEOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx)) to all the display areas;
      * the read attributes access right ([DESKTOP\_READOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx) ) to all the display areas;
      * the display area creation access right ([WINSTA\_CREATEDESKTOP](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the display area enumeration access right ([WINSTA\_ENUMDESKTOPS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the clipboard access right to all the display areas ([WINSTA\_ACCESSCLIPBOARD](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS graphic device interface (GDI) Manager (win32k.sys) operations:
      * the screen content access right to all the display areas ([WINSTA\_READSCREEN](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS kernel executive manager (ex.lib) operations (e.g. registering a character string to the global atom table maintained by the Windows OS kernel and receiving a unique value [called an atom for identifying the registered string]):
      * the global atom manipulation access right ([WINSTA\_ACCESSGLOBALATOMS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx)) to all the display areas;
  + the instance of the window logon state maintaining service must have reset the security of the user interactive application display area so that no subject of any user account is granted the permission for accessing the user interactive application display area.

### “Hibernate during locked” state

The instance of the window logon state maintaining service requests the Local Window Terminal Service (LWTS) (lsm.exe) to disconnect the window terminal session where the instance of the window logon state maintaining service resides in, upon entering the “Hibernate during locked” state, in the remote terminal session case.

The instance of the window logon state maintaining service reacts to one of the following notifications, upon entering the “Hibernate during locked” state.

* A notification originated from the Windows OS power manager or another authorized source to abort the entering of the shutdown, hibernation, or standing-by power state.
* A notification originated from the Windows OS power manager or another authorized source to resume from the hibernation or the standing-by power state and to request the window logon user interface service (aka LogonUI.exe) to display the below messages.
  + “Windows is resuming...” from the last hibernation.
  + “Windows is resuming...” from the last “standing by” system power state.

While still remaining in the “Hibernate during locked” state, if the instance of the window logon state maintaining service requests the window logon user interface service (aka LogonUI.exe) to display resume messages on the secure display area, then

* there must have been a notification originated from the Windows OS power manager or another authorized source to resume from the hibernation or the standing-by power state and to request the window logon user interface service (aka LogonUI.exe) to display the below messages.
  + “Windows is resuming...” from the last hibernation.
  + “Windows is resuming...” from the last “standing by” system power state.

### Transitions from the “Hibernate during locked” state

The “Hibernate during locked” state supports the following state transitions.

* If the “Hibernate during locked” state transitions to the “Locked” state, then
  + there must have been a notification originated from the Windows OS power manager or another authorized source to abort the entering of the shutdown, hibernation, or standing-by power state.

### “Attempt to unlock” state

The instance of the window logon state maintaining service requests the local Windows OS Authentication Service (aka lsass.exe) to authenticate the interactive user using his supplied credential information (such as the user name with a password or the smart card PIN) corresponding to a credential provider installed on the local machine and to confirm that the identity of this user being authenticated in this current attempt to unlock matches the identity of the authenticated user who has logged on, upon entering the “Attempt to unlock” state.

### Transitions from the “Attempt to unlock” state

The “Attempt to unlock” state supports the following state transitions.

* If the “Attempt to unlock” state transitions to the “Report unlock success” state, then
  + the local Windows OS Authentication Service (aka lsass.exe) must have reported success for the credential information supplied by the interactive user;
  + the identity of this user having been authenticated in this current attempt to unlock must have matched the identity of the authenticated user who has logged on;
  + the instance of the window logon state maintaining service must have retrieved the information about the authenticated user’s previous logons, in the case where the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy is enabled.
* If the “Attempt to unlock” state transitions to the “Report unlock failure” state, then either:
  + the local Windows OS Authentication Service (aka lsass.exe) must have reported a failure for the credential information supplied by the interactive user;
  + the identity of this user having been authenticated in this current attempt to unlock must not have matched the identity of the authenticated user who has logged on;
  + the instance of the window logon state maintaining service must not have retrieved the information about the authenticated user’s previous logons, in the case where the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy is enabled.

### “Report unlock success” state

The instance of the window logon state maintaining service conducts the following:

* generates the [Event ID 4801](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_WORKSTATION\_UNLOCKED) “The (workstation) display areas are unlocked” audit record in the hard audit store;
* queries the administrator specified “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy,

upon entering the “Report unlock success” state.

### Transitions from the “Report unlock success” state

The “Report unlock success” state supports the following state transitions.

* If the “Report unlock success” state transitions to the “Report last logon to user after unlock” state, then
  + the queried “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy must have been enabled;
  + the successful user authentication for this current unlock attempt must not have been based on cached credential related data.
* If the “Report unlock success” state transitions to the “User logged on” state, then either:
  + the queried “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy must not have been enabled;
  + the successful user authentication for this current unlock attempt must have been based on cached credential related data.

### “Report unlock failure” state

The instance of the window logon state maintaining service requests the window logon user interface service to query the responsible credential provider (which has been used to collect the user credential information) to translate the specified logon/authentication status code reported by the Windows OS Authentication Service (aka lsass.exe) to the appropriate user-readable message, upon entering the “Report unlock failure” state.

In the local terminal session case, if the number of consecutive failed authentication occurrences upon retrying to authenticate as the same user account during an unlock attempt exceeds the service-defined consecutive failed authentication count (namely the LOCKOUT\_BAD\_LOGON\_COUNT (5) within the LOCKOUT\_BAD\_LOGON\_PERIOD (60 seconds)), then

* the instance of the window logon state maintaining service introduces a service-defined delay period (in seconds) in its processing.

Namely, the instance of the window logon state maintaining service introduces the LOCKOUT\_BAD\_LOGON\_DELAY (30 seconds) in its processing by sending the Windows OS to sleep for the LOCKOUT\_BAD\_LOGON\_DELAY (30 seconds), as described in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.3” requirement.

The instance of the window logon state maintaining service requests the window logon user interface service to display the specified error message on the secure display area based on the logon/authentication status reported by the Windows OS Authentication Service and mediated by the Window Logon State Maintaining Service, and to retrieve the user’s response, upon entering the “Report unlock failure” state.

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports a logon failure status ([STATUS\_LOGON\_FAILURE](http://msdn.microsoft.com/en-us/library/cc704588.aspx)) or a wrong password status ([STATUS\_WRONG\_PASSWORD](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “The user name or password is incorrect.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the user account is not allowed to logon at the current time due to the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy assigned to the account by an administrator ([STATUS\_INVALID\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “Your account has time restrictions that prevent you from logging on at this time.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the user account is disabled ([STATUS\_ACCOUNT\_DISABLED](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “Your account has been disabled.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the user account is expired ([STATUS\_ACCOUNT\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “Your account has expired.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the user account’s password has expired and changing the password is necessary ([STATUS\_PASSWORD\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “The password for this account has expired.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the user account has been configured by the administrator that it can only conduct an interactive logon using a smart card ([STATUS\_SMARTCARD\_LOGON\_REQUIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “You must use a smart card to log on to or unlock this computer.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the user account has been automatically locked out because too many invalid password based logon attempts or password change attempts have been requested ([STATUS\_ACCOUNT\_LOCKED\_OUT](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “The user account has been automatically locked out because too many invalid password based logon attempts or password change attempts have been requested.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that no Windows OS domain controllers are not available ([STATUS\_NO\_LOGON\_SERVERS](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “The domain specified is not available. Please try again later.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the local machine cannot establish a trust relationship with a Windows OS domain controller that the local machine trusts ([STATUS\_TRUSTED\_RELATIONSHIP\_FAILURE](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “Windows cannot connect to the domain, either because the domain controller is down or otherwise unavailable, or because your computer account was not found. Please try again later. If this message continues to appear, contact your system administrator for assistance.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the PIN supplied by the smart card interactive user for logging is incorrect ([STATUS\_SMARTCARD\_WRONG\_PIN](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “An incorrect PIN was presented to the smart card.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the smart card does not contain the required logon certificate ([STATUS\_SMARTCARD\_NO\_CERTIFICATE](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “The requested certificate does not exist on the smart card.”

The window logon user interface service displays one of the following messages on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the smart card does not contain the key set ([STATUS\_SMARTCARD\_NO\_KEYSET](http://msdn.microsoft.com/en-us/library/cc704588.aspx)) or key container ([STATUS\_SMARTCARD\_NO\_KEY\_CONTAINER](http://msdn.microsoft.com/en-us/library/cc704588.aspx)) corresponding to the required logon certificate.

* “The requested keyset does not exist on the smart card.”
* “The requested key container does not exist on the smart card.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the smart card certificate used for the logon authentication has expired ([STATUS\_SMARTCARD\_CERT\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “The smart card certificate used for the logon authentication has expired.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the smart card certificate used for the logon authentication has been revoked ([STATUS\_SMARTCARD\_CERT\_REVOKED](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “The smart card certificate used for the logon authentication has been revoked.”

The window logon user interface service displays one of the following messages on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that an untrusted certificate authority was detected while processing the smart card certificate used for the logon authentication ([STATUS\_ISSUING\_CA\_UNTRUSTED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) or [STATUS\_REVOCATION\_OFFLINE\_KDC](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “An untrusted certificate authority was detected while processing the smart card certificate used for the logon authentication.”
* “The revocation status of the domain controller certificate used for smart card authentication could not be determined. There is additional information in the system event log. Please contact your system administrator.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the revocation status of the smart card certificate used for the logon authentication could not be determined because the required revocation server is not available ([STATUS\_REVOCATION\_OFFLINE\_C](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “The revocation status of the smart card certificate used for authentication could not be determined.”

The window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that there is a failure in the verification of the user’s smart card public key cryptography based credential.

* “Your credentials could not be verified.”

The window logon user interface service keeps the specified error message as being displayed on the secure display area until the interactive user selects one of the following as his response, or until the dialog timeout occurs.

* “OK”;
* “Cancel”,

as presented on the secure display area.

### Transitions from the “Report unlock failure” state

The “Report unlock failure” state supports the following state transitions.

* If the “Report unlock failure” state transitions to the “Request to unlock” state so that the interactive user can retry to supply his/her credential information again for another unlock attempt, then
  + the logon or authentication status reported by the Windows OS Authentication Service must not have been that of one of the following:
    - the user account’s password has expired and changing the password is necessary ([STATUS\_PASSWORD\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the user account has been automatically locked out because too many invalid password based logon attempts or password change attempts have been requested ([STATUS\_ACCOUNT\_LOCKED\_OUT](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the user account is not allowed to logon at the current time due to the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy assigned to the account by an administrator ([STATUS\_INVALID\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the user account is disabled ([STATUS\_ACCOUNT\_DISABLED](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the user account is expired ([STATUS\_ACCOUNT\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the revocation status of the smart card certificate used for the logon authentication could not be determined because the required revocation server is not available ([STATUS\_REVOCATION\_OFFLINE\_C](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the smart card certificate used for the logon authentication has expired ([STATUS\_SMARTCARD\_CERT\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the smart card does not contain the required logon certificate ([STATUS\_SMARTCARD\_NO\_CERTIFICATE](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the smart card does not contain the key set ([STATUS\_SMARTCARD\_NO\_KEYSET](http://msdn.microsoft.com/en-us/library/cc704588.aspx)) or key container ([STATUS\_SMARTCARD\_NO\_KEY\_CONTAINER](http://msdn.microsoft.com/en-us/library/cc704588.aspx)) corresponding to the required logon certificate;
    - an untrusted certificate authority was detected while processing the smart card certificate used for the logon authentication ([STATUS\_ISSUING\_CA\_UNTRUSTED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) or [STATUS\_REVOCATION\_OFFLINE\_KDC](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).
* If the “Report unlock failure” state transitions to the “Locked” state, then either:
  + the logon or authentication status reported by the Windows OS Authentication Service must have been that of one of the following:
    - the user account’s password has expired and changing the password is necessary ([STATUS\_PASSWORD\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the user account has been automatically locked out because too many invalid password based logon attempts or password change attempts have been requested ([STATUS\_ACCOUNT\_LOCKED\_OUT](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the user account is not allowed to logon at the current time due to the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy assigned to the account by an administrator ([STATUS\_INVALID\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the user account is disabled ([STATUS\_ACCOUNT\_DISABLED](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the user account is expired ([STATUS\_ACCOUNT\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the revocation status of the smart card certificate used for the logon authentication could not be determined because the required revocation server is not available ([STATUS\_REVOCATION\_OFFLINE\_C](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the smart card certificate used for the logon authentication has expired ([STATUS\_SMARTCARD\_CERT\_EXPIRED](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the smart card does not contain the required logon certificate ([STATUS\_SMARTCARD\_NO\_CERTIFICATE](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
    - the smart card does not contain the key set ([STATUS\_SMARTCARD\_NO\_KEYSET](http://msdn.microsoft.com/en-us/library/cc704588.aspx)) or key container ([STATUS\_SMARTCARD\_NO\_KEY\_CONTAINER](http://msdn.microsoft.com/en-us/library/cc704588.aspx)) corresponding to the required logon certificate;
    - an untrusted certificate authority was detected while processing the smart card certificate used for the logon authentication ([STATUS\_ISSUING\_CA\_UNTRUSTED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) or [STATUS\_REVOCATION\_OFFLINE\_KDC](http://msdn.microsoft.com/en-us/library/cc704588.aspx));
  + there must have been a timeout for receiving the user response through a window dialog displayed by the window logon user interface service.
* If the “Report unlock failure” state transitions to the “Locked” state and then to the “Session disconnected during locked” state[[21]](#footnote-22), then
  + the Local Window Terminal Service (LWTS) (lsm.exe) must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in.
* If the “Report unlock failure” state transitions to the “Locked” state and then to the “Hibernate during locked” state[[22]](#footnote-23), then
  + there must have been a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service to display the below messages:
    - “Windows is shutting down...”;
    - “Undocking and going to sleep...”;
    - “Hibernating...”;
    - “Going to sleep...”, corresponding to the entering of the “standing by” Windows OS system power state.
* If the “Report unlock failure” state transitions to the “Welcome” state, then
  + either:
    - the local Windows OS shutdown initiation service (as part of wininit.exe) must have notified a logoff or shutdown;
    - a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + the instance of the window logon state maintaining service must have switched to the secure display area;
  + the instance of the window logon state maintaining service must have terminated any remaining applications (if any) which belong to the authenticated interactive user account;
  + the instance of the window logon state maintaining service generates the [Event ID 4647](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_BEGIN\_LOGOFF.id) “User initiated logoff” audit record in the hard audit store;
  + the instance of the window logon state maintaining service must have requested its local Windows OS window manager to assign no one (i.e. the null [LUID](http://msdn.microsoft.com/en-us/library/aa379261.aspx)) as the interactive logged on user of the window manager’s corresponding window terminal session;
  + the instance of the window logon state maintaining service must have revoked any of the following rights to the display areas (residing in the window terminal session where the service resides in) that may have been granted to the previously logged on user account:
    - the specific rights that are necessary in some of the Windows OS window manager operations:
      * the write attributes access right ([DESKTOP\_WRITEOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx)) to all the display areas;
      * the read attributes access right ([DESKTOP\_READOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx) ) to all the display areas;
      * the display area creation access right ([WINSTA\_CREATEDESKTOP](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the display area enumeration access right ([WINSTA\_ENUMDESKTOPS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the clipboard access right to all the display areas ([WINSTA\_ACCESSCLIPBOARD](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS graphic device interface (GDI) Manager (win32k.sys) operations:
      * the screen content access right to all the display areas ([WINSTA\_READSCREEN](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS kernel executive manager (ex.lib) operations (e.g. registering a character string to the global atom table maintained by the Windows OS kernel and receiving a unique value [called an atom for identifying the registered string]):
      * the global atom manipulation access right ([WINSTA\_ACCESSGLOBALATOMS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx)) to all the display areas;
  + the instance of the window logon state maintaining service must have reset the security of the user interactive application display area so that no subject of any user account is granted the permission for accessing the user interactive application display area.

### “Report last logon to user after unlock” state

The instance of the window logon state maintaining service requests the window logon user interface service to display messages related to the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy and to collect the interactive user’s response, upon entering the “Report last logon to user after unlock” state.

The window logon user interface service displays one of the following message box messages on the secure display area to collect the interactive user’s response after the success authentication of the interactive user for this current unlock attempt, in the case where the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy is enabled.

* “<user account name> Successful Logon: The last time you interactively logged on to this account was: <Date/Time of the previous success logon>. Unsuccessful Logon: There have been no unsuccessful interactive logon attempts with this account since your last interactive logon.”
* “<user account name> Successful Logon: The last time you interactively logged on to this account was: <Date/Time of the previous success logon>. Unsuccessful Logon: The last unsuccessful interactive logon attempt on this account was: <Date/Time of the previous failed logon>. The number of unsuccessful interactive logon attempts since your last interactive logon: <failed attempt count>.”
* “Security policies on this computer are set to display information about the last interactive logon. However, this information could not be retrieved. ”

In the message box containing any of the above messages about the information of previous logons, the window logon user interface service provides only the “OK” button for collecting the interactive user’s response.

The window logon user interface service keeps any of the above messages about the information of previous logons as being displayed on the secure display area until the interactive user selects one of the following as his/her response, or until the dialog timeout occurs.

* “OK”,

as presented on the secure display area.

### Transitions from the “Report last logon to user after unlock” state

The “Report last logon to user after unlock” state supports the following state transitions.

* If the “Report last logon to user after unlock” state transitions to the “User logged on” state, then
  + there must have been the interactive user’s acknowledgement (in the form of clicking the “OK” button in the message box containing one of the above messages about the information of previous logons) as his/her response to the displayed message about his/her previous logons on the secure display area;
  + the instance of the window logon state maintaining service must have switched back to the user interactive application display area.
* If the “Report last logon to user after unlock” state transitions to the “Locked” state, then either:
  + there must have been a timeout for receiving the user response through a window dialog displayed by the window logon user interface service;
  + there must have been a failure reported from the window logon user interface service during the display of the message about the information of previous logons.
* If the “Report last logon to user after unlock” state transitions to the “Locked” state and then to the “Session disconnected during locked” state[[23]](#footnote-24), then
  + the Local Window Terminal Service (LWTS) (lsm.exe) must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in.
* If the “Report last logon to user after unlock” state transitions to the “Locked” state and then to the “Hibernate during locked” state[[24]](#footnote-25), then
  + there must have been a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service to display the below messages:
    - “Windows is shutting down...”;
    - “Undocking and going to sleep...”;
    - “Hibernating...”;
    - “Going to sleep...”, corresponding to the entering of the “standing by” Windows OS system power state.
* If the “Report last logon to user after unlock” state transitions to the “Welcome” state, then
  + either:
    - the local Windows OS shutdown initiation service (as part of wininit.exe) must have notified a logoff or shutdown;
    - a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + the instance of the window logon state maintaining service must have switched to the secure display area;
  + the instance of the window logon state maintaining service must have terminated any remaining applications (if any) which belong to the authenticated interactive user account;
  + the instance of the window logon state maintaining service generates the [Event ID 4647](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_BEGIN\_LOGOFF.id) “User initiated logoff” audit record in the hard audit store;
  + the instance of the window logon state maintaining service must have requested its local Windows OS window manager to assign no one (i.e. the null [LUID](http://msdn.microsoft.com/en-us/library/aa379261.aspx)) as the interactive logged on user of the window manager’s corresponding window terminal session;
  + the instance of the window logon state maintaining service must have revoked any of the following rights to the display areas (residing in the window terminal session where the service resides in) that may have been granted to the previously logged on user account:
    - the specific rights that are necessary in some of the Windows OS window manager operations:
      * the write attributes access right ([DESKTOP\_WRITEOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx)) to all the display areas;
      * the read attributes access right ([DESKTOP\_READOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx) ) to all the display areas;
      * the display area creation access right ([WINSTA\_CREATEDESKTOP](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the display area enumeration access right ([WINSTA\_ENUMDESKTOPS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the clipboard access right to all the display areas ([WINSTA\_ACCESSCLIPBOARD](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS graphic device interface (GDI) Manager (win32k.sys) operations:
      * the screen content access right to all the display areas ([WINSTA\_READSCREEN](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS kernel executive manager (ex.lib) operations (e.g. registering a character string to the global atom table maintained by the Windows OS kernel and receiving a unique value [called an atom for identifying the registered string]):
      * the global atom manipulation access right ([WINSTA\_ACCESSGLOBALATOMS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx)) to all the display areas;
  + the instance of the window logon state maintaining service must have reset the security of the user interactive application display area so that no subject of any user account is granted the permission for accessing the user interactive application display area.

### Summary of the Windows OS justification for meeting “3.2.1.2”

Because of the availability of the “lock the display areas from visibility” option for the interactive user to select in the “Trusted path activated” state, the Windows OS clearly supports the “user-initiated locking of an interactive session” functionality specified in this “3.2.1.2” requirement.

Due to the switching to the secure display area before entering the “Locked” state, the user interactive application display area, belonging to the instance of the window logon state maintaining service, is not visible and is not receiving user input. Therefore, the Windows OS accomplishes the following actions of this “3.2.1.2” requirement.

1. Clearing or overwriting display devices, making the current contents unreadable;
2. Disabling any activity from the user’s data access/display devices other than unlocking the interactive session.

Additionally, from the above description of the transitions from the relevant states, it is clear that any transition leading to an entering of the “User logged on” states requires a successful attempt of user authentication, when the following relevant administrator specified policies are followed as explained in the above transition descriptions.

* The “[display area locking](http://technet2.microsoft.com/windowsserver/en/library/c7410c7f-d6cf-400b-9604-f1e2b4fe1d2c1033.mspx?mfr=true)” policy;
* The “[disable trusted path](http://technet2.microsoft.com/windowsserver/en/library/83e7e44d-84fe-4e8b-9e5c-df97609eb8601033.mspx?mfr=true)” policy;
* The “[Hide entry points for Fast User Switching](http://support.microsoft.com/kb/329885)” policy;
* The “User input inactivity timeout” policy;
* The “Screen saver” policy.

## Addressing 3.2.1.3 “The OS shall require a specific set of actions from users and administrators”

The Commercial Grade OS Requirement Set requires the following specific set of actions from users and administrators:

1. A user (needs) to re-authenticate to unlock an interactive session;
2. An administrator (needs) to authenticate before the system security functions unlock and automatically terminate a user interactive session.

This requirement is addressed by the Windows OS as follows.

Action a) is obvious in the Windows OS. It is seen in the “Request to unlock” state as explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.2.1.2” requirement. Additionally, from the description of the transitions from the relevant states, given in the justification text for addressing the Commercial Grade OS Requirement Set “3.2.1.2” requirement, it is clear that any transition leading to an entering of the “User logged on” states requires a successful attempt of user authentication, when the following relevant administrator specified policies are followed as explained in the above transition descriptions.

* The “[display area locking](http://technet2.microsoft.com/windowsserver/en/library/c7410c7f-d6cf-400b-9604-f1e2b4fe1d2c1033.mspx?mfr=true)” policy;
* The “[disable trusted path](http://technet2.microsoft.com/windowsserver/en/library/83e7e44d-84fe-4e8b-9e5c-df97609eb8601033.mspx?mfr=true)” policy;
* The “[Hide entry points for Fast User Switching](http://support.microsoft.com/kb/329885)” policy;
* The “User input inactivity timeout” policy;
* The “Screen saver” policy.

Action b) seems to be a side-effect consequence of a “single user only” operating system. This action is handled differently on the Windows OS. Except the Windows Vista Starter SKU, all Windows Vista SKUs support the “[Fast User Switching](http://windowshelp.microsoft.com/Windows/en-US/Help/4b2208e2-90ac-4394-b7fc-b8a84b2e2d0a1033.mspx)” functionality. With this “[Fast User Switching](http://windowshelp.microsoft.com/Windows/en-US/Help/4b2208e2-90ac-4394-b7fc-b8a84b2e2d0a1033.mspx)” functionality, a window terminal session does not need to be terminated in order to let another user logon to a different window terminal session of the same machine. The current local window terminal session merely needs to be disconnected from the local display terminal for displaying the contents (i.e. window graphical interfaces) of its display areas and accepting user input, so that another window terminal session is connected to the local display terminal instead for displaying the contents (i.e. window graphical interfaces) of its display areas and accepting user input. The new current local window terminal session becomes available to any interactive user, not just to an administrator.

The below subsections explains the behaviors in more detail.

### “Switch user” button

Let us denote the current local window terminal session, connected to the local display terminal for displaying the contents (i.e. window graphical interfaces) of its display areas and accepting user input, as Session X.

In the local window terminal session case, as explained in the ““Locked” state” section of this paper, the instance of the window logon state maintaining service in Session X requests the window logon user interface service to display the following “Switch user” option on the secure display area through the “Switch user” button, upon entering the “Locked” state.

* An option to switch to another local window terminal session to allow a different user to log on or switch to another logon user account’s local or remote window terminal session, in the case where the session switch behavior is allowed by the administrator specified “[Hide entry points for Fast User Switching](http://support.microsoft.com/kb/329885)” policy.

If the interactive user selects the “Switch user” option, then

* the window logon user interface service makes a request to the Local Window Terminal Service (LWTS) (lsm.exe) (for which the window logon user interface service should have the necessary permissions in the [WTSDisconnectSession()](http://msdn.microsoft.com/en-us/library/aa383830.aspx) interface of lsm.exe) to disconnect the currently visible window terminal session, namely Session X, from the local display terminal.

According to the corresponding security functional assertion of Local Window Terminal Service (LWTS), as stated in the [Microsoft publication: “Security Functional Assertions of the “User Interaction based on Windowing” Scenario of a Modern Operating System”](http://download.microsoft.com/download/e/d/b/edbb17fb-580d-49a4-b66c-8726cf446a86/User%20Interaction.doc), the other effects are that:

* the Local Window Terminal Service (LWTS) generates the [[Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record which identifies Session X after the Session X disconnection occurs;
* a new local window terminal session, say Session Y, is created;
* an instance of the Windows OS window manager (namely win32k.sys) for Session Y starts;
* an instance of the window logon state maintaining service (winlogon.exe) for Session Y starts;
* an instance of the window logon user interface service (aka LogonUI.exe) for Session Y starts;
* Session Y, as the new local window terminal session, is connected to the local display terminal so that the “Welcome” message from the Session Y window logon user interface service instance is displayed to the local interactive user in the “Welcome” state of the Session Y window logon state maintaining service instance;
* the Local Window Terminal Service (LWTS) generates the [[Event ID 4778](http://support.microsoft.com/kb/947226) “A session was reconnected to a Window Station” (SE\_AUDITID\_ETW\_SESSION\_RECONNECTED\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record which identifies Session Y after the Session Y reconnection occurs.

Following the transitions from the “Welcome” state of the Session Y window logon state maintaining service instance, the interactive user needs to supply his/her credential information for the local Windows OS Authentication Service (aka lsass.exe) to authenticate the interactive user. The ““Window logon state maintaining service user interactive application display area initialization” phase” in Session Y (as the new local window terminal session) is carried out only if the interactive user authentication is successful.

### “Window logon state maintaining service user interactive application display area initialization” phase

As described in the [Microsoft publication: “Security Functional Assertions of the “User Interaction based on Windowing” Scenario of a Modern Operating System”](http://download.microsoft.com/download/e/d/b/edbb17fb-580d-49a4-b66c-8726cf446a86/User%20Interaction.doc), every instance of the window logon state maintaining service (aka winlogon.exe), including that of Session Y, contains, among other states:

* the “Report logon service OK to logon UI service”;
* the “Report last logon to user”;
* the “Get session confirmation from LWTS” state;
* the “Notify logon subscribers” state;
* the “Starting shell” state;
* the “User logged on” state,

in the “Window logon state maintaining service user interactive application display area initialization” phrase, which is depicted in the following diagram.



### “Report logon service OK to logon UI service” state

The “Report logon service OK to logon UI service” state can only be entered after

* the local Windows OS Authentication Service (aka lsass.exe) has reported success for the credential information supplied by the interactive user;
* the authenticated interactive user is confirmed as an administrator, in the case where the hard audit storage is currently full;
* the instance of the window logon state maintaining service has retrieved the information about the authenticated user’s previous logons, in the case where the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy is enabled.

The instance of the window logon state maintaining service queries the administrator specified “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy, upon entering the “Report logon service OK to logon UI service” state.

### Transitions from the “Report logon service OK to logon UI service” state

The “Report logon service OK to logon UI service” state supports the following state transitions.

* If the “Report logon service OK to logon UI service” state transitions to the “Report last logon to user” state, then
  + the queried “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy must have been enabled;
  + the successful logon authentication must not have been based on cached credential related data.
* If the “Report logon service OK to logon UI service” state transitions to the “Get session confirmation from LWTS” state, then either:
  + the queried “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy must not have been enabled;
  + the successful logon authentication must have been based on cached credential related data;
  + the Local Window Terminal Service must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in;
  + the local shutdown initiation service (as part of ininit.exe) must have notified a logoff or shutdown;
  + a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service.

### “Report last logon to user” state

The instance of the window logon state maintaining service requests the window logon user interface service to display messages related to the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy and to collect the interactive user’s response, upon entering the “Report last logon to user” state.

The window logon user interface service displays one of the following message box messages on the secure display area to collect the interactive user’s response after the success authentication of the interactive user for this current interactive logon attempt, in the case where the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy is enabled.

* “<user account name> This is the first time you have interactively logged on to this account.”
* “<user account name> This is the first time you have interactively logged on to this account. Unsuccessful Logon: The last unsuccessful interactive logon attempt on this account was: <Date/Time of the previous failed logon>. The number of unsuccessful interactive logon attempts since your account was created: <failed attempt count>.”
* “<user account name> Successful Logon: The last time you interactively logged on to this account was: <Date/Time of the previous success logon>. Unsuccessful Logon: There have been no unsuccessful interactive logon attempts with this account since your last interactive logon.”
* “<user account name> Successful Logon: The last time you interactively logged on to this account was: <Date/Time of the previous success logon>. Unsuccessful Logon: The last unsuccessful interactive logon attempt on this account was: <Date/Time of the previous failed logon>. The number of unsuccessful interactive logon attempts since your last interactive logon: <failed attempt count>.”
* “Security policies on this computer are set to display information about the last interactive logon. However, this information could not be retrieved. ”

In the message box containing any of the above messages about the information of previous logons, the window logon user interface service provides only the “OK” button for collecting the interactive user’s response.

The window logon user interface service keeps any of the above messages about the information of previous logons as being displayed on the secure display area until the interactive user selects one of the following as his/her response, or until the dialog timeout occurs.

* “OK”,

as presented on the secure display area.

### Transitions from the “Report last logon to user” state

The “Report last logon to user” state supports the following state transitions.

* If the “Report last logon to user” state transitions to the “Get session confirmation from LWTS” state, then
  + there must have been the interactive user’s acknowledgement (in the form of clicking the “OK” button in the message box containing one of the above messages about the information of previous logons) as his/her response to the displayed message about his/her previous logons on the secure display area.
* If the “Report last logon to user” state transitions to the “Welcome” state, then either:
  + the Local Window Terminal Service (LWTS) (lsm.exe) must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in;
  + the local Windows OS shutdown initiation service (as part of wininit.exe) must have notified a logoff or shutdown;
  + a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + there must have been a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service to display the below messages:
    - “Windows is shutting down...”;
    - “Undocking and going to sleep...”;
    - “Hibernating...”;
    - “Going to sleep...”, corresponding to the entering of the “standing by” Windows OS system power state.
  + there must have been a timeout for receiving the user response through a window dialog displayed by the window logon user interface service;
  + there must have been a notification of an authorized subject’s initiation of a locking of the display areas or there must have been a request originated from an authorized subject to lock the display areas;
  + there must have been a failure reported from the window logon user interface service during the display of the message about the information of previous logons.
* If the “Report last logon to user” state transitions to the “Welcome” state, then
  + the instance of the window logon state maintaining service must have requested its local Windows OS window manager to assign no one (i.e. the null [LUID](http://msdn.microsoft.com/en-us/library/aa379261.aspx)) as the interactive logged on user of the window manager’s corresponding window terminal session;
  + the instance of the window logon state maintaining service must have revoked any of the following rights to the display areas (residing in the window terminal session where the service resides in) that may have been granted to the previously logged on user account:
    - the specific rights that are necessary in some of the Windows OS window manager operations:
      * the write attributes access right ([DESKTOP\_WRITEOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx)) to all the display areas;
      * the read attributes access right ([DESKTOP\_READOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx) ) to all the display areas;
      * the display area creation access right ([WINSTA\_CREATEDESKTOP](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the display area enumeration access right ([WINSTA\_ENUMDESKTOPS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the clipboard access right to all the display areas ([WINSTA\_ACCESSCLIPBOARD](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS graphic device interface (GDI) Manager (win32k.sys) operations:
      * the screen content access right to all the display areas ([WINSTA\_READSCREEN](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS kernel executive manager (ex.lib) operations (e.g. registering a character string to the global atom table maintained by the Windows OS kernel and receiving a unique value [called an atom for identifying the registered string]):
      * the global atom manipulation access right ([WINSTA\_ACCESSGLOBALATOMS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx)) to all the display areas;
  + the instance of the window logon state maintaining service must have reset the security of the user interactive application display area so that no subject of any user account is granted the permission for accessing the user interactive application display area.

### “Get session confirmation from LWTS” state

The instance of the window logon state maintaining service requests the Local Window Terminal Service (LWTS) to use one of its arbitrators to determine whether the current window terminal session, namely Session Y (as the new local window terminal session), should be continued for completing the logon procedures of the already authenticated interactive user, upon entering the “Get session confirmation from LWTS” state.

There could be multiple requests to the Local Window Terminal Service (LWTS) for its final determination while the instance of the window logon state maintaining service is remaining in the “Get session confirmation from LWTS” state.

#### “Select one of the active but disconnected sessions to reconnect to” dialog

While still remaining in the “Get session confirmation from LWTS” state, if the instance of the window logon state maintaining service conducts the following:

* requests the window logon user interface service to display the “select one of the active but disconnected sessions to reconnect to” dialog on the secure display area for retrieving the interactive user’s response which captures his/her desired session selection;
* submits the interactive user’s desired session selection to the Local Window Terminal Service (LWTS) for further approval determination,

then

* the instance of the window logon state maintaining service must have received a determination from the Local Window Terminal Service (LWTS) that the instance of the window logon state maintaining service requests the window logon user interface service to display the “select one of the active but disconnected sessions to reconnect to” dialog on the secure display area for retrieving the interactive user’s response which captures his/her desired session selection.

When the Local Window Terminal Service (LWTS) determinates that the instance of the window logon state maintaining service requests the window logon user interface service to display the “select one of the active but disconnected sessions to reconnect to” dialog on the secure display area, the LWTS provides a list of active but disconnected window terminal sessions for the interactive user to select. The LWTS arbitrators ensure that the authenticated interactive user actually belongs to the active but disconnected window terminal sessions for selection.

The “select one of the active but disconnected sessions to reconnect to” dialog contains the following message items:

* “Select a session to reconnect to.”
  + <elapsed length of session logged on time> and <elapsed length of session disconnected time> to identify the corresponding window terminal session for selection.

The window logon user interface service keeps the “select one of the active but disconnected sessions to reconnect to” dialog message as being displayed on the secure display area until the interactive user selects a displayed session as his response, or until the interactive user cancels the dialog, or until the dialog timeout occurs.

#### Dialogs available to an administrator

While still remaining in the “Get session confirmation from LWTS” state, if the instance of the window logon state maintaining service conducts the following:

* requests the window logon user interface service to display one of the following dialogs on the secure display area for retrieving the interactive user’s response which captures his/her desired action or session selection:
  + the “end another session” dialog;
  + the “disconnect another session” dialog;
  + the “select one of the sessions to force disconnection” dialog;
  + the “select one of the sessions to request disconnection from the selected session’s logged on user” dialog;
* submits the interactive user’s desired action or session selection to the Local Window Terminal Service (LWTS) for further approval determination,

then

* the instance of the window logon state maintaining service must have received a determination from the Local Window Terminal Service (LWTS) that the instance of the window logon state maintaining service requests the window logon user interface service to display one of the above dialogs on the secure display area for retrieving the interactive user’s response which captures his/her desired action or session selection.

When the Local Window Terminal Service (LWTS) determinates that the instance of the window logon state maintaining service requests the window logon user interface service to display one of the following dialogs on the secure display area,

* + the “end another session” dialog;
  + the “disconnect another session” dialog;
  + the “select one of the sessions to force disconnection” dialog;
  + the “select one of the sessions to request disconnection from the selected session’s logged on user” dialog,

the LWTS arbitrators ensure that the authenticated interactive user actually is an administrator.

##### The “end another session” dialog

The “end another session” dialog contains the following message items:

* “Another user is currently logged on to this computer. If you continue, this user's window terminal session will end and any un-saved data will be lost. Do you want to continue?”
  + “yes” and “no” buttons for capturing the interactive user’s response.

The window logon user interface service keeps the “end another session” dialog message as being displayed on the secure display area until the interactive user selects one of the following as his response, or until the dialog timeout occurs:

* “yes”;
* “no”,

as presented on the secure display area.

##### “Disconnect another session” dialog

The “disconnect another session” dialog contains the following message items:

* “Another user is currently logged on to this computer. If you continue, this user has to disconnect his window terminal session from this computer. Do you want to continue?”
  + “yes” and “no” buttons for capturing the interactive user’s response.

The window logon user interface service keeps the “disconnect another session” dialog message as being displayed on the secure display area until the interactive user selects one of the following as his response, or until the dialog timeout occurs:

* “yes”;
* “no”,

as presented on the secure display area.

##### “Select one of the sessions to force disconnection” dialog

The “select one of the sessions to force disconnection” dialog contains the following message items:

* “Select a user to disconnect so that you may logon. Select one of the below users to force disconnection.”
  + <user name> is away and his window terminal session is locked;
  + <user name> is active and is working on his window terminal session;
  + <user name> is idle for <number of seconds> on his window terminal session.

The window logon user interface service keeps the “select one of the sessions to force disconnection” dialog message as being displayed on the secure display area until the interactive user selects a displayed session as his response, or until the interactive user cancels the dialog, or until the dialog timeout occurs.

##### “Select one of the sessions to request disconnection from the selected session’s logged on user” dialog

The “select one of the sessions to request disconnection from the selected session’s logged on user” dialog contains the following message items:

* “Select a user to request disconnection so that you may logon. There are too many users logged in.”
  + <user name> is away and his window terminal session is locked;
  + <user name> is active and is working on his window terminal session;
  + <user name> is idle for <number of seconds> on his window terminal session.

The window logon user interface service keeps the “select one of the sessions to request disconnection from the selected session’s logged on user” dialog message as being displayed on the secure display area until the interactive user selects a displayed session as his response, or until the interactive user cancels the dialog, or until the dialog timeout occurs.

#### Display the “<user name> denied your disconnect request” message

While still remaining in the “Get session confirmation from LWTS” state, if the instance of the window logon state maintaining service conducts the following:

* requests the window logon user interface service to display the “<user name> denied your disconnect request” message on the secure display area for retrieving the interactive user’s response which captures his/her desired action, where
  + the <user name> corresponds to the user chosen by the interactive user in the previously displayed “select one of the sessions to request disconnection from the selected session’s logged on user” dialog on the secure display area;
* submits the interactive user’s desired action or session selection to the Local Window Terminal Service (LWTS) for further approval determination,

then

* the instance of the window logon state maintaining service must have received a determination from the Local Window Terminal Service (LWTS) that the instance of the window logon state maintaining service requests the window logon user interface service to display the “<user name> denied your disconnect request” message on the secure display area for retrieving the interactive user’s response which captures his/her desired action.

The window logon user interface service keeps the “<user name> denied your disconnect request” message as being displayed on the secure display area until the interactive user selects one of the following as his/her response, or until the dialog timeout occurs.

* “OK”,

as presented on the secure display area.

### Transitions from the “Get session confirmation from LWTS” state

The “Get session confirmation from LWTS” state supports the following state transitions.

* If the “Get session confirmation from LWTS” state transitions to the “Notify logon subscribers” state, then
  + the instance of the window logon state maintaining service must have received the final determination from the Local Window Terminal Service (LWTS) that the current window terminal session, namely Session Y (as the new local window terminal session), is continued for completing the logon procedures of the already authenticated interactive user.
* If the “Get session confirmation from LWTS” state transitions to the “Welcome” state, then either:
  + the Local Window Terminal Service (LWTS) (lsm.exe) must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in;
  + the local Windows OS shutdown initiation service (as part of wininit.exe) must have notified a logoff or shutdown;
  + a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + there must have been a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service to display the below messages:
    - “Windows is shutting down...”;
    - “Undocking and going to sleep...”;
    - “Hibernating...”;
    - “Going to sleep...”, corresponding to the entering of the “standing by” Windows OS system power state.
  + there must have been a timeout for receiving the user response through a window dialog displayed by the window logon user interface service;
  + there must have been a notification of an authorized subject’s initiation of a locking of the display areas or there must have been a request originated from an authorized subject to lock the display areas;
  + the instance of the window logon state maintaining service must have received the final determination from the Local Window Terminal Service that the current window terminal session, namely Session Y (as the new local window terminal session), is terminated so that the logon procedures of the already authenticated interactive user are aborted for the current window terminal session.
* If the “Get session confirmation from LWTS” state transitions to the “Welcome” state, then
  + the instance of the window logon state maintaining service must have requested its local Windows OS window manager to assign no one (i.e. the null [LUID](http://msdn.microsoft.com/en-us/library/aa379261.aspx)) as the interactive logged on user of the window manager’s corresponding window terminal session;
  + the instance of the window logon state maintaining service must have revoked any of the following rights to the display areas (residing in the window terminal session where the service resides in) that may have been granted to the previously logged on user account:
    - the specific rights that are necessary in some of the Windows OS window manager operations:
      * the write attributes access right ([DESKTOP\_WRITEOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx)) to all the display areas;
      * the read attributes access right ([DESKTOP\_READOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx) ) to all the display areas;
      * the display area creation access right ([WINSTA\_CREATEDESKTOP](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the display area enumeration access right ([WINSTA\_ENUMDESKTOPS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the clipboard access right to all the display areas ([WINSTA\_ACCESSCLIPBOARD](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS graphic device interface (GDI) Manager (win32k.sys) operations:
      * the screen content access right to all the display areas ([WINSTA\_READSCREEN](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS kernel executive manager (ex.lib) operations (e.g. registering a character string to the global atom table maintained by the Windows OS kernel and receiving a unique value [called an atom for identifying the registered string]):
      * the global atom manipulation access right ([WINSTA\_ACCESSGLOBALATOMS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx)) to all the display areas;
  + the instance of the window logon state maintaining service must have reset the security of the user interactive application display area so that no subject of any user account is granted the permission for accessing the user interactive application display area.
* If the “Get session confirmation from LWTS” state transitions to the “Welcome” state and this instance of the window logon state maintaining service (of Session Y, as the new local window terminal session) sends a notification to another instance of the window logon state maintaining service running in a different source window terminal session (say Session S) to provide the update of the below user characteristics of the already authenticated interactive user upon reconnection (of Session S where the receiving instance of the window logon state maintaining service resides in) with the information freshly obtained by the local Windows OS Authentication Service:
  + the time when the user should log off subject to the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy;
  + the time and date the password was last changed;
  + the time and date when the user should be reminded to change the account logon password;
  + the time and date when the user must change the account logon password;
  + other user account attributes such as:
    - the user authentication in the current logon attempt was based on cached credential related data;
    - the account that the user was logged on is the guest account;
    - the user was authenticated using a clear text password stored in the Windows OS domain controllers;
    - the user was authenticated using a password hash stored in the Windows OS domain controllers;
  + the time and date when the user was authenticated by the local Windows OS Authentication Service;
  + a request to reset the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy warning timer so that a warning reminder about the action associated with the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy would be displayed to the logged on interactive user when the warning timer expires;
  + a request to reset the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy action timer so that the action associated with the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy would be carried out by Session S window logon state maintaining service when the action timer expires,

then

* + the Session Y window logon state maintaining service must have received the final determination from the Local Window Terminal Service (LWTS) that:
    - the current window terminal session (namely Session Y, as the new local window terminal session) is disconnected from the local display terminal;
    - Session Y is terminated so that the logon procedures of the already authenticated interactive user are aborted for Session Y (as the new local window terminal session);
    - the already authenticated interactive user has been determined that s/he belongs to Session S;
    - Session S is a disconnected window terminal session or a remote connected window terminal session;
    - the Local Window Terminal Service (LWTS) connects Session S to the local display terminal so that contents (i.e. window graphical interfaces) from the display areas of Session S are displayed to the local interactive user (who is the already authenticated interactive user) and input from the local interactive user is accepted;
  + in the case where Session S is a disconnected window terminal session:
    - the Local Window Terminal Service (LWTS) requests Session Y’s Windows OS window manager (win32k.sys) to disconnect Session Y from the local display terminal so that no contents (i.e. window graphical interfaces) from the display areas of Session S are displayed to the local interactive user (who is the already authenticated interactive user) and no input from the local interactive user is accepted;
    - the Local Window Terminal Service (LWTS) generates the [[Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record which identifies Session Y after the Session Y disconnection occurs;
    - the Local Window Terminal Service requests Session S’s Windows OS window manager (win32k.sys) to connect Session S to the local display terminal so that contents (i.e. window graphical interfaces) from Session S are displayed to the local interactive user, who is the already authenticated interactive user previously seeing window graphical interfaces from Session Y, and input from the local interactive user is accepted;
    - the Local Window Terminal Service (LWTS) generates the [[Event ID 4778](http://support.microsoft.com/kb/947226) “A session was reconnected to a Window Station” (SE\_AUDITID\_ETW\_SESSION\_RECONNECTED\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record which identifies Session S after the Session S reconnection occurs;
    - Session S becomes the current local connected window terminal session instead of Session Y;
    - Session Y becomes a disconnected window terminal session;
  + in the case where Session S is a remote connected window terminal session having an active network stack (say Stack S) to a remote network RDP client:
    - the Windows OS remote window terminal service (termsrv.dll) requests Session S’s Windows OS window manager (win32k.sys) to disconnect Session S from Stack S so that no contents (i.e. window graphical interfaces) from Session S are displayed to the remote interactive user of the remote network RDP client and no input from the remote interactive user is accepted;
    - the Windows OS remote window terminal service (termsrv.dll) notifies the remote network RDP client about the termination of the client’s connection to Session S;
    - Stack S for communicating with the remote network RDP client is closed;
    - the Windows OS remote window terminal service (termsrv.dll) generates the [[Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record which identifies Session S after the Session S disconnection occurs;
    - the Windows OS remote window terminal service (termsrv.dll) requests Session Y’s Windows OS window manager (win32k.sys) to disconnect Session Y from the local display terminal so that no contents (i.e. window graphical interfaces) from Session Y are displayed to the local interactive user and no input from local interactive user is accepted;
    - the Windows OS remote window terminal service (termsrv.dll) generates the [[Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record which identifies Session Y after the Session Y disconnection occurs;
    - the Windows OS remote window terminal service (termsrv.dll) requests Session S’s Windows OS window manager (win32k.sys) to connect Session S to the local display terminal so that contents (i.e. window graphical interfaces) from Session S are displayed to the local interactive user, who is the already authenticated interactive user previously seeing window graphical interfaces from Session Y, and input from the local interactive user is accepted;
    - the Windows OS remote window terminal service (termsrv.dll) generates the [[Event ID 4778](http://support.microsoft.com/kb/947226) “A session was reconnected to a Window Station” (SE\_AUDITID\_ETW\_SESSION\_RECONNECTED\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record which identifies Session S after the Session S reconnection occurs;
    - Session S becomes a local connected window terminal session;
    - Session Y becomes a disconnected window terminal session.

### “Notify logon subscribers” state

The instance of the window logon state maintaining service conducts the following, upon entering the “Notify logon subscribers” state.

* The instance of the window logon state maintaining service requests its local Windows OS window manager to assign the already authenticated user as the interactive logon user of the window manager’s corresponding window terminal session.
* The instance of the window logon state maintaining service grants the already authenticated user the permission (for accessing the user interactive application display area) so that the user possesses the following rights to the display areas (residing in the window terminal session where the service resides in):
  + the specific rights that are necessary in some of the Windows OS window manager operations:
    - the write attributes access right ([DESKTOP\_WRITEOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx)) to all the display areas;
    - the read attributes access right ([DESKTOP\_READOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx) ) to all the display areas;
    - the display area creation access right ([WINSTA\_CREATEDESKTOP](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the display area enumeration access right ([WINSTA\_ENUMDESKTOPS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the clipboard access right to all the display areas ([WINSTA\_ACCESSCLIPBOARD](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
  + the specific rights that are necessary in certain Windows OS graphic device interface (GDI) Manager (win32k.sys) operations:
    - the screen content access right to all the display areas ([WINSTA\_READSCREEN](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
  + the specific rights that are necessary in certain Windows OS kernel executive manager (ex.lib) operations (e.g. registering a character string to the global atom table maintained by the Windows OS kernel and receiving a unique value [called an atom for identifying the registered string]):
    - the global atom manipulation access right ([WINSTA\_ACCESSGLOBALATOMS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx)) to all the display areas;
* The instance of the window logon state maintaining service resets the security of the user interactive application display area so that the already authenticated user is granted the permission for accessing the user interactive application display area.

The instance of the window logon state maintaining service conducts the following, upon entering the “Notify logon subscribers” state.

* The instance of the window logon state maintaining service grants the already authenticated user the permission (for accessing the user interactive application display area) so that the user possesses the following rights to the nameable and securable object container associated with the window terminal session where the service resides in:
  + read security control (READ\_CONTROL);
  + object and sub-container query (DIRECTORY\_QUERY);
  + object or sub-container name lookup (DIRECTORY\_TRAVERSE);
  + object name creation (DIRECTORY\_CREATE\_OBJECT);
  + sub-container creation (DIRECTORY\_CREATE\_SUBDIRECTORY).

The nameable and securable object container of a specific window terminal session is a default object container which contains nameable and securable objects, such as events, local process call (LPC) ports, symbolic links, mutexes, and others, which are created by subjects residing in the specific window terminal session. Having a window terminal session specific nameable and securable object container supports better isolation of window terminal session. This object container is the “\Sessions\<Session ID>\BaseNamedObjects” directory with the <Session ID> variable being the actual Session ID of the window terminal session where this instance of the window logon state maintaining service resides in.

The instance of the window logon state maintaining service conducts the following, upon entering the “Notify logon subscribers” state.

* The enforcement of the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy for the already authenticated interactive user who is not an administrator.

The possible action values which could be used to determine the action (which is the action to be taken by the instance of the window logon state maintaining service when the user’s allowable logon hours expire) due to the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy are:

* “nothing”;
* lock the display areas;
* disconnect the window terminal session where the instance of the window logon state maintaining service resides in;
* forced logoff for the already authenticated interactive user.

While still remaining in the “Notify logon subscribers” state, if the instance of the window logon state maintaining service receives a failure in its attempt to connect to an administrator-specified logon notify subscriber, then

* the instance of the window logon state maintaining service initiates a forced logoff for the already authenticated interactive user who is not an administrator.

While still remaining in the “Notify logon subscribers” state, if the instance of the window logon state maintaining service fails to notify an administrator-specified logon notify subscriber about the current interactive logon, then

* the instance of the window logon state maintaining service initiates a forced logoff for the already authenticated interactive user.

While still remaining in the “Notify logon subscribers” state, if the already authenticated user’s allowable logon hours have expired in the case where the action value of the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy is not “nothing”, then

* the instance of the window logon state maintaining service initiates a forced logoff for the already authenticated interactive user who is not an administrator.

While still remaining in the “Notify logon subscribers” state, if the action for the already authenticated interactive user who is not an administrator is due to a non-“nothing” action of the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy, then

* the instance of the window logon state maintaining service generates the [Event ID 4002](http://technet.microsoft.com/en-us/library/cc733989.aspx) “The logon hours restriction policy is applied to the logged on user. The user's session has been locked, disconnected or logged off depending on the policy setting” (EVENT\_LOGON\_HOURS\_POLICY\_APPLIED) soft audit record which identifies the already authenticated interactive user for the soft audit storage.

The soft audit storage is the audit storage that does not cause the local machine to prevent subsequent auditable events, except those taken by an administrator, to occur when the audit storage is full (i.e. reaching its maximum size).

While still remaining in the “Notify logon subscribers” state, if the Local Window Terminal Service has notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in, then

* the instance of the window logon state maintaining service initiates a forced logoff for the already authenticated interactive user who is not an administrator.

### Transitions from the “Notify logon subscribers” state

The “Notify logon subscribers” state supports the following state transitions.

* If the “Notify logon subscribers” state transitions to the “Starting shell” state, then
  + the instance of the window logon state maintaining service must have notified all administrator-specified logon notify subscribers successfully;
  + the already authenticated interactive user must have been allowed by the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy.
* If the “Notify logon subscribers” state transitions to the “Welcome” state, then
  + either:
    - the local Windows OS shutdown initiation service (as part of wininit.exe) must have notified a logoff or shutdown;
    - a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + the instance of the window logon state maintaining service must have switched to the secure display area;
  + the instance of the window logon state maintaining service must have terminated any remaining applications (if any) which belong to the authenticated interactive user account;
  + the instance of the window logon state maintaining service generates the [Event ID 4647](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_BEGIN\_LOGOFF.id) “User initiated logoff” audit record in the hard audit store;
  + the instance of the window logon state maintaining service must have requested its local Windows OS window manager to assign no one (i.e. the null [LUID](http://msdn.microsoft.com/en-us/library/aa379261.aspx)) as the interactive logged on user of the window manager’s corresponding window terminal session;
  + the instance of the window logon state maintaining service must have revoked any of the following rights to the display areas (residing in the window terminal session where the service resides in) that may have been granted to the previously logged on user account:
    - the specific rights that are necessary in some of the Windows OS window manager operations:
      * the write attributes access right ([DESKTOP\_WRITEOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx)) to all the display areas;
      * the read attributes access right ([DESKTOP\_READOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx) ) to all the display areas;
      * the display area creation access right ([WINSTA\_CREATEDESKTOP](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the display area enumeration access right ([WINSTA\_ENUMDESKTOPS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the clipboard access right to all the display areas ([WINSTA\_ACCESSCLIPBOARD](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS graphic device interface (GDI) Manager (win32k.sys) operations:
      * the screen content access right to all the display areas ([WINSTA\_READSCREEN](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS kernel executive manager (ex.lib) operations (e.g. registering a character string to the global atom table maintained by the Windows OS kernel and receiving a unique value [called an atom for identifying the registered string]):
      * the global atom manipulation access right ([WINSTA\_ACCESSGLOBALATOMS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx)) to all the display areas;
  + the instance of the window logon state maintaining service must have reset the security of the user interactive application display area so that no subject of any user account is granted the permission for accessing the user interactive application display area.

### “Starting shell” state

The instance of the window logon state maintaining service conducts the following, upon entering the “Starting shell” state.

* The instance of the window logon state maintaining service switches to the user interactive application display area.
* The instance of the window logon state maintaining service initializes the user environment for the already authenticated user on the user interactive application display area, where the user environment initialization includes:
  + the Shell program for the user;
  + the desk window manager service (UxSms.dll and aka: “User Experience Session Management Service”) to manage the user application window objects residing on the user interactive application display area;
  + the administrator specified logon script for the user.

While still remaining in the “Starting shell” state, if the instance of the window logon state maintaining service initiates a forced logoff for the already authenticated interactive user, then

* the already authenticated user must not be an administrator;
* the user environment initialization (including:
  + the Shell program for the user;
  + the desk window manager service to manage the user application window objects residing on the user interactive application display area;
  + the administrator specified logon script for the user);

must have been failed.

### Transitions from the “Starting shell” state

The “Starting shell” state supports the following state transitions.

* If the “Starting shell” state transitions to the “User logged on” state, then either:
  + the already authenticated user must be an administrator
  + the user environment initialization (including:
    - the Shell program for the user;
    - the desk window manager service to manage the user application window objects residing on the user interactive application display area;
    - the administrator specified logon script for the user);

must have been completed.

* If the “Starting shell” state transitions to the “Welcome” state, then
  + either:
    - the local Windows OS shutdown initiation service (as part of wininit.exe) must have notified a logoff or shutdown;
    - a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + the instance of the window logon state maintaining service must have switched to the secure display area;
  + the instance of the window logon state maintaining service must have terminated any remaining applications (if any) which belong to the authenticated interactive user account;
  + the instance of the window logon state maintaining service generates the [Event ID 4647](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_BEGIN\_LOGOFF.id) “User initiated logoff” audit record in the hard audit store;
  + the instance of the window logon state maintaining service must have requested its local Windows OS window manager to assign no one (i.e. the null [LUID](http://msdn.microsoft.com/en-us/library/aa379261.aspx)) as the interactive logged on user of the window manager’s corresponding window terminal session;
  + the instance of the window logon state maintaining service must have revoked any of the following rights to the display areas (residing in the window terminal session where the service resides in) that may have been granted to the previously logged on user account:
    - the specific rights that are necessary in some of the Windows OS window manager operations:
      * the write attributes access right ([DESKTOP\_WRITEOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx)) to all the display areas;
      * the read attributes access right ([DESKTOP\_READOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx) ) to all the display areas;
      * the display area creation access right ([WINSTA\_CREATEDESKTOP](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the display area enumeration access right ([WINSTA\_ENUMDESKTOPS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the clipboard access right to all the display areas ([WINSTA\_ACCESSCLIPBOARD](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS graphic device interface (GDI) Manager (win32k.sys) operations:
      * the screen content access right to all the display areas ([WINSTA\_READSCREEN](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS kernel executive manager (ex.lib) operations (e.g. registering a character string to the global atom table maintained by the Windows OS kernel and receiving a unique value [called an atom for identifying the registered string]):
      * the global atom manipulation access right ([WINSTA\_ACCESSGLOBALATOMS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx)) to all the display areas;
  + the instance of the window logon state maintaining service must have reset the security of the user interactive application display area so that no subject of any user account is granted the permission for accessing the user interactive application display area.

### “User logged on” state

The instance of the window logon state maintaining service reacts to one of the following notifications while the already authenticated interactive user has his/her interactive application display area visible and ready to receive user input, upon entering the “User logged on” state.

* The local Windows OS shutdown initiation service (as part of wininit.exe) notifies a logoff or shutdown.
* A logoff or shutdown is triggered internally within the instance of the window logon state maintaining service.
* There is a notification of an authorized subject’s initiation of a locking of the display areas or there is a request originated from an authorized subject to lock the display areas.
* There is a notification of an arrival of the registered display area locking activation request hot key (e.g. Windows Key+L in the default situation) which has been mediated by the local Windows OS window manager of the same window terminal session.
* There is a notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session.
* There is a notification of a trusted path simulation from an authorized subject.
* There is a notification of an arrival of the registered task manager (taskmgr.exe) activation request hot key (e.g. CTRL+SHIFT+ESC in the default situation) which has been mediated by the local Windows OS window manager of the same window terminal session.
* The Local Window Terminal Service notifies disconnection of the window terminal session where the instance of the window logon state maintaining service resides in.
* There is a notification from the another instance of the window logon state maintaining service running in a different window terminal session to provide the update of the characteristics of the currently logged on user upon reconnection (of the window terminal session where the receiving instance of the window logon state maintaining service resides in) with the information freshly obtained by the local Windows OS Authentication Service.
* There is a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service (aka LogonUI.exe) to display the below messages:
  + “Windows is shutting down...” corresponding to data saving;
  + “Undocking and going to sleep...”;
  + “Hibernating...”
  + “Going to sleep...” corresponding to the entering of the “standing by” Windows OS system power state;
* There is a request for the instance of the window logon state maintaining service to switch the current active display area (of the window terminal session where the service resides in) to the secure display area for displaying the user consent dialog window (created by the Administrative Privileged Application Launching Service [consent.exe]) with the slightly-faded background window which is a snapshot of the entire application display area that would have been taken after the instance of the window logon state maintaining service has confirmed that the current active display area is the user interactive application display area.
* There is a notification of the user input inactivity timeout which has been mediated by the local Windows OS window manager of the same window terminal session.

While still remaining in the “User logged on” state, if the instance of the window logon state maintaining service conducts the following:

* updates the below user characteristics of the already authenticated interactive user upon reconnection with the information freshly obtained by the local Windows OS Authentication Service:
  + the time when the user should log off subject to the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy;
  + the time and date the password was last changed;
  + the time and date when the user should be reminded to change the account logon password;
  + the time and date when the user must change the account logon password;
  + other user account attributes such as:
    - the user authentication in the current logon attempt was based on cached credential related data;
    - the account that the user was logged on is the guest account;
    - the user was authenticated using a clear text password stored in the Windows OS domain controllers;
    - the user was authenticated using a password hash stored in the Windows OS domain controllers;
  + the time and date when the user was authenticated by the local Windows OS Authentication Service;
* enforces the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy for the authenticated interactive user;
* displays the “hard audit storage is full” message on the user interactive application display area in the case where the already authenticated interactive user is an administrator and the hard audit storage is full (i.e. reaching its maximum size);
* displays the “your authentication was based on cached credential related data” message on the user interactive application display area in the case where the already authenticated interactive user was authenticated based on cached credential related data and the administrator specified “report cached logon to the interactive logged on user” policy is enabled,

then

* there must have been a notification from the another instance of the window logon state maintaining service running in a different window terminal session to provide the update of the characteristics of the currently logged on user upon reconnection (of the window terminal session where the receiving instance of the window logon state maintaining service resides in) with the information freshly obtained by the local Windows OS Authentication Service.

If enabled, the “report cached logon to the interactive logged on user” policy specifies that the instance of the window logon state maintaining service would report to the interactive logged user in the case where the logon server was not available during the user’s interactive logging on processing.

The following registry key values defining the “report cached logon to the interactive logged on user” policy value do not allow the non privileged user to write.

* HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\ReportControllerMissing;
* HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System\ReportControllerMissing;
* HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System\ReportControllerMissing.

While still remaining in the “User logged on” state, if the instance of the window logon state maintaining service conducts the following:

* initiates the locking of the display areas for the already authenticated interactive user who is not an administrator, in the case where the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy action value is “lock the display areas”;
* requests the Local Window Terminal Service (LWTS) to disconnect the window terminal session where the instance of the window logon state maintaining service resides in for the already authenticated interactive user who is not an administrator, in the case where the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy action value is “disconnect the window terminal session where the instance of the window logon state maintaining service resides in”;
* initiates a forced logoff for the already authenticated interactive user who is not an administrator, in the case where the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy action value is “forced logoff”,

then

* the already authenticated user’s allowable logon hours must have expired in the case where the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy is not “nothing”.

While still remaining in the “User logged on” state, if the instance of the window logon state maintaining service conducts the following:

* starts the task manager program (taskmgr.exe) as the authenticated interactive user to run on the user interactive application display area,

then

* there must have been a notification of an arrival of the registered task manager activation request hot key (e.g. CTRL+SHIFT+ESC in the default situation) which has been mediated by the local Windows OS window manager of the same window terminal session;
* the “display task manager” policy must have been enabled.

If a subject requests to set the value which is used to determine whether the “display task manager” policy is enabled or not, as a per user behavior item for the user account that the subject represents, then

* the behavior item must be allowed by the administrator specified policy.

The following registry key values defining the “display task manager” policy value do not allow the non privileged user to write.

* HKEY\_CURRENT\_USER\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\DisableTaskMgr;
* HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System\DisableTaskMgr.

While still remaining in the “User logged on” state, if the instance of the window logon state maintaining service conducts the following:

* displays the “You are about to be logged off” message or the “Logoff is cancelled” message on the user interactive application display area,

then

* there must have been a notification from the local Windows OS shutdown initiation service (as part of wininit.exe) informing the instance of the window logon state maintaining service to remind the currently logged on interactive user on the window terminal session (where the instance of the window logon state maintaining service resides in) about either:
  + the imminent forced log off;
  + the log off cancellation.

### Transitions from the “User logged on” state

The “User logged on” state supports the following state transitions.

* If the “User logged on” state transitions to the “Session disconnected” state” state, then
  + the Local Window Terminal Service (LWTS) must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in.
* If the “User logged on” state transitions to the “User consent dialog” state, then

there must have been a request for the instance of the window logon state maintaining service to switch the current active display area (of the window terminal session where the service resides in) to the secure display area for displaying the user consent dialog window (created by the Administrative Privileged Application Launching Service

* + Appendix A: Administrative Privileged Application Launching Service [consent.exe]) with the slightly-faded background window which is a snapshot of the entire application display area that would have been taken after the instance of the window logon state maintaining service has confirmed that the current active display area is the user interactive application display area.
* If the “User logged on” state transitions to the “Post logged on inactivity timeout handler” state, then
  + there must have been a notification of the user input inactivity timeout which has been mediated by the local Windows OS window manager of the same window terminal session.
* If the “User logged on” state transitions to the “Trusted path activated” state, then
  + there must have been either:
    - the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
    - the notification of a trusted path simulation from an authorized subject.
* If the “User logged on” state transitions to the “Initiate lock on display areas” state, then
  + there must have been either:
    - a notification of an authorized subject’s initiation of a locking of the display areas or a request originated from an authorized subject to lock the display areas;
    - a notification of an arrival of the registered display area locking activation request hot key (e.g. Windows Key+L in the default situation) which has been mediated by the local Windows OS window manager of the same window terminal session.
* If the “User logged on” state transitions to the “Post logged on hibernate” state, then
  + there must have been a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service (aka LogonUI.exe) to display the below messages:
    - “Windows is shutting down...” corresponding to data saving;
    - “Undocking and going to sleep...”;
    - “Hibernating...”
    - “Going to sleep...” corresponding to the entering of the “standing by” Windows OS system power state.
* If the “User logged on” state transitions to the “Welcome” state, then
  + either:
    - the local Windows OS shutdown initiation service (as part of wininit.exe) must have notified a logoff or shutdown;
    - a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + the instance of the window logon state maintaining service must have switched to the secure display area;
  + the instance of the window logon state maintaining service must have terminated any remaining applications (if any) which belong to the authenticated interactive user account;
  + the instance of the window logon state maintaining service generates the [Event ID 4647](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_BEGIN\_LOGOFF.id) “User initiated logoff” audit record in the hard audit store;
  + the instance of the window logon state maintaining service must have requested its local Windows OS window manager to assign no one (i.e. the null [LUID](http://msdn.microsoft.com/en-us/library/aa379261.aspx)) as the interactive logged on user of the window manager’s corresponding window terminal session;
  + the instance of the window logon state maintaining service must have revoked any of the following rights to the display areas (residing in the window terminal session where the service resides in) that may have been granted to the previously logged on user account:
    - the specific rights that are necessary in some of the Windows OS window manager operations:
      * the write attributes access right ([DESKTOP\_WRITEOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx)) to all the display areas;
      * the read attributes access right ([DESKTOP\_READOBJECTS](http://msdn.microsoft.com/en-us/library/ms682575(vs.85).aspx) ) to all the display areas;
      * the display area creation access right ([WINSTA\_CREATEDESKTOP](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the display area enumeration access right ([WINSTA\_ENUMDESKTOPS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
      * the clipboard access right to all the display areas ([WINSTA\_ACCESSCLIPBOARD](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS graphic device interface (GDI) Manager (win32k.sys) operations:
      * the screen content access right to all the display areas ([WINSTA\_READSCREEN](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx));
    - the specific rights that are necessary in certain Windows OS kernel executive manager (ex.lib) operations (e.g. registering a character string to the global atom table maintained by the Windows OS kernel and receiving a unique value [called an atom for identifying the registered string]):
      * the global atom manipulation access right ([WINSTA\_ACCESSGLOBALATOMS](http://msdn.microsoft.com/en-us/library/ms687391(VS.85).aspx)) to all the display areas;
  + the instance of the window logon state maintaining service must have reset the security of the user interactive application display area so that no subject of any user account is granted the permission for accessing the user interactive application display area.

### Summary of the Windows OS justification for meeting the requirement for an administrator to unlock and terminate a user session

Recall that Action b) of the Commercial Grade OS Requirement Set “3.2.1.3” requirement states that “an administrator (needs) to authenticate before the system security functions unlock and automatically terminate a user interactive session”. However, due to the “Fast User Switching” functionality of the Windows OS and the following dialogs available to an administrator during the “Get session confirmation from LWTS” state, the Action b) requirement, as stated, is moot.

* the “end another session” dialog;
* the “disconnect another session” dialog;
* the “select one of the sessions to force disconnection” dialog;
* the “select one of the sessions to request disconnection from the selected session’s logged on user” dialog.

Additionally, from the description of all the transitions from the “Get session confirmation from LWTS” state, it is clear that the reachable states collectively support the intents behind the Action b) requirement without destroying the “work” belonging to an active, but locked, window terminal session of a user unnecessarily.

Consequently, this Commercial Grade OS Requirement Set “3.2.1.3” requirement is addressed.

## Addressing 3.2.1.4 “The OS shall enforce an authorized administrator specified maximum number of concurrent interactive sessions per user”

This requirement is addressed by the Windows OS as follows.

In a distributed environment where the Windows OS machines typically are deployed, it is unrealistic to assume a high reliability level of the underlying (wired or wireless) network transport medium in the environment. Indeed, the Windows OS provides the ability to optionally allow a user to attempt a cached logging on to a user account as explained in the “Logging on attempt through a cached password verifier” section of this paper. In this situation, the user at least can grant access to the authorized resources residing in the local Windows OS machine, where the cached logging on takes place, even though networked resources residing in remote Windows OS machines are momentarily unavailable due to a network transport interruption.

Furthermore, the Windows OS domain controllers (and hence the Windows OS Active Directory) also cannot receive a user account logging off notification reliably as a high reliability level of the underlying (wired or wireless) network transport cannot be guaranteed. As a result, the Windows OS domain controllers do not collectively have a reliable mechanism to keep track of the current number of interactive logging on sessions occurring within the Windows OS forest/domain for a specific user account. For the majority of machines connected to the network, physically unplugging the network cable from the machine or turning off the machine integrated wireless network interface card (NIC) is relatively easy. Unfortunately, there is no effective logical countermeasure to the risk of physically tempering of the network cable or the machine integrated wireless NIC. At the same time, typical users also require a certain level of mobility for their laptop or tablet PC machines. Most physical solutions to counter the risk of physically tempering of the network cable or the machine integrated wireless NIC are impractical from the mobility perspective. Consequently, the centralized enforcement of a maximum number of interactive logging on sessions for a specific user account introduces a hard-to-counter denial of service (DOS) attack to the specific user account. We believe that the elimination of this hard-to-counter denial of service (DOS) attack outweighs the usefulness of the centralized enforcement of a maximum number of interactive logging on sessions for a specific user account.

Additionally, a Kerberos or NLTM client connected to the Windows OS forest/domain may be implemented by a 3rd party. There is no guarantee that the 3rd party Kerberos or NLTM client always provide the Windows OS domain controllers a user account logging off notification.

While the above have reasoned that keeping track of the current number of interactive logging on sessions occurring within the Windows OS forest/domain for a specific user account is impractical, the below presents a “workaround” solution for meeting the intent of this requirement.

We recall the following Windows OS system access rights that may be assigned to a user account or a group in a local Windows OS machine in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.8” requirement:

* [SeInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx);
* [SeDenyInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx);
* [SeRemoteInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx);
* [SeDenyRemoteInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx).

The “workaround” solution relies on the enforcement of these Windows OS system access rights for a specific user account locally on individual Windows OS machines. Hence, there is no dependency on the network transport reliability at all times.

In the “workaround” solution, we let N be the maximum number of interactive logging on sessions for a specific user account defined by an administrator. The administrator selects N Windows OS machines which are member computers of the Windows OS forest/domain managed by the administrator.

In each of those N selected Windows OS machines, the administrator

* assigns the [SeInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx) and [SeRemoteInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx) Windows OS system access rights to the specific user account.

In each of the remaining Windows OS machines in the Windows OS forest/domain which are not those N selected Windows OS machines, the administrator

* assigns the [SeDenyInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx) and [SeDenyRemoteInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx) Windows OS system access rights to the specific user account.

The enforcement effect of the above Windows OS system access right assignment to the specific user account clearly implies that N is the maximum number of interactive logging on sessions for the specific user account.

Finally, we note that the Windows OS system access right assignment on individual Windows OS machines is just part of the Windows OS group policy based security settings. As explained in the justification text for addressing the Commercial Grade OS Requirement Set “1.2.2.1” requirement, the Windows OS group policy based security settings for individual Windows OS machines can be managed and configured centralized by an administrator in his/her Windows OS forest/domain.

Consequently, this Commercial Grade OS Requirement Set “3.2.1.4” requirement is addressed.

## Addressing 3.2.1.5 “The OS shall lock an interactive session after an authorized administrator specified time interval of user inactivity by performing a specific set of actions”

The Commercial Grade OS Requirement Set requires the following specific set of actions being performed by the OS:

1. Clearing or overwriting display devices, making the current contents unreadable;
2. Disabling any interactive user activity from that display devices other than unlocking the interactive session.

This requirement is addressed by the Windows OS as follows.

As described in the [Microsoft publication: “Security Functional Assertions of the “User Interaction based on Windowing” Scenario of a Modern Operating System”](http://download.microsoft.com/download/e/d/b/edbb17fb-580d-49a4-b66c-8726cf446a86/User%20Interaction.doc), every instance of the window logon state maintaining service (aka winlogon.exe) contains, among other states:

* the “Inactivity timeout to run screensaver” state;
* the “Post logged on inactivity timeout handler” state;
* the “Post logged on resume” state;
* the “Initiate lock during resume” state;
* the “Inactivity timeout handler during locked” state.

These states are transitioned to after the arrival of a notification of the user input inactivity timeout which has been mediated by the local Windows OS window manager of the same window terminal session.

### “Inactivity timeout to run screensaver” state

The “Inactivity timeout to run screensaver” state is transitioned from the “Welcome” state after the arrival of a notification of the user input inactivity timeout which has been mediated by the local Windows OS window manager of the same window terminal session.

The instance of the window logon state maintaining service conducts the following:

* switches to the screen saver display area;
* starts the screen saver application, assigned in the administrator-specified “screen saver” policy, in a Windows OS process to run on the screen saver display area,

in the case where

* the physical video monitor is not currently turned off;
* the “user input inactivity timeout” policy specifies “Start the screen saver” or “Start a password protected screen saver”,

upon entering the “Inactivity timeout to run screensaver” state.

While still remaining in the “Inactivity timeout to run screensaver” state, the instance of the window logon state maintaining service waits for the termination of the administrator specified screen saver application (according to the “screen saver” policy) started by the service.

The administrator specified screen saver application process started by the instance of the window logon state maintaining service terminates itself upon receiving the following window messages from the local Windows OS window manager:

* a mouse button is physically pressed ([WM\_LBUTTONDOWN](http://msdn.microsoft.com/en-us/library/ms645607.aspx), [WM\_MBUTTONDOWN](http://msdn.microsoft.com/en-us/library/ms645610(VS.85).aspx), [WM\_RBUTTONDOWN](http://msdn.microsoft.com/en-us/library/ms646242(VS.85).aspx));
* the mouse is physically moved by a specific threshold distance ([WM\_MOUSEMOVE](http://msdn.microsoft.com/en-us/library/ms645616(VS.85).aspx));
* a non-system keyboard key is pressed, where a non-system key is a keyboard key that is pressed when the ALT key is not pressed ([WM\_KEYDOWN](http://msdn.microsoft.com/en-us/library/ms646280(VS.85).aspx));
* the translated character of the non-system key being pressed ([WM\_CHAR](http://msdn.microsoft.com/en-us/library/ms646276(VS.85).aspx)).

The instance of the window logon state maintaining service switches back to the secure display area before exiting from the “Inactivity timeout to run screensaver” state.

### Transitions from the “Inactivity timeout to run screensaver” state

The “Inactivity timeout to run screensaver” state supports the following state transitions.

* If the “Inactivity timeout to run screensaver” state transitions to the “Welcome” state, then either:
  + administrator specified screen saver application process started by the instance of the window logon state maintaining service must have terminated;
  + the Local Window Terminal Service must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in;
  + the local shutdown initiation service (as part of ininit.exe) must have notified a logoff or shutdown;
  + a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + there must have been either:
    - the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
    - the notification of a trusted path simulation from an authorized subject;
  + there must have been a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service (aka LogonUI.exe) to display the below messages:
    - “Windows is shutting down...” corresponding to data saving;
    - “Undocking and going to sleep...”;
    - “Hibernating...”
    - “Going to sleep...” corresponding to the entering of the “standing by” Windows OS system power state.

### “Post logged on inactivity timeout handler” state

The “Post logged on inactivity timeout handler” state is transitioned from the following states:

* the “User logged on” state;
* the “Credential backup” state;
* the “Credential restore” state;
* the “Prompting for credentials” state;
* the “Change passwords” state,

after the arrival of a notification of the user input inactivity timeout which has been mediated by the local Windows OS window manager of the same window terminal session.

The instance of the window logon state maintaining service conducts the following:

* switches to the screen saver display area;
* starts the default screen saver application running in the security context of the authenticated interactive user on the screen saver display area,

upon entering the “Post logged on inactivity timeout handler” state, in the case where either:

* + the physical video monitor is not currently turned off;
  + the “user input inactivity timeout” policy specifies “Start a password protected screen saver”;
  + the screen saver grace period is non-zero;
  + the “user input inactivity timeout” policy specifies “Locking of the display areas”;
  + the screen saver grace period is non-zero;
  + the “user input inactivity timeout” policy specifies “Log off”;
  + the screen saver grace period is non-zero.

While still remaining in the “Post logged on inactivity timeout handler” state, the instance of the window logon state maintaining service waits for the elapse of the screen saver grace period before the default screen saver application process started by the instance of the window logon state maintaining service terminates.

The instance of the window logon state maintaining service conducts the following:

* switches to the screen saver display area;
* starts the administrator specified screen saver application (according to the “screen saver” policy) running in the security context of the authenticated interactive user on the screen saver display area;
* generates the [Event ID 4802](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_SCREENSAVER\_INVOKED) “The screen saver was invoked” audit record in the hard audit store,

upon entering the “Post logged on inactivity timeout handler” state, in the case where either:

* + the “user input inactivity timeout” policy specifies “Start a password protected screen saver”;
  + the “user input inactivity timeout” policy specifies “Start the screen saver”;
  + the “screen saver” policy specifies “the screen saver application process runs on its own screen saver display area”.

The instance of the window logon state maintaining service conducts the following:

* starts the administrator specified screen saver application (according to the “screen saver” policy) running in the security context of the authenticated interactive user on the user interactive application display area;
* generates the [Event ID 4802](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_SCREENSAVER\_INVOKED) “The screen saver was invoked” audit record in the hard audit store,

upon entering the “Post logged on inactivity timeout handler” state, in the case where

* the “user input inactivity timeout” policy specifies “Start the screen saver”;
* the “screen saver” policy specifies “the screen saver application process runs on the user interactive application display area”.

While still remaining in the “Post logged on inactivity timeout handler” state, the instance of the window logon state maintaining service waits for the termination of the administrator specified screen saver application (according to the “screen saver” policy) started by the service.

The instance of the window logon state maintaining service conducts the following:

* initiates a locking of the display areas;
* switches to the secure display area,

before exiting from the “Post logged on inactivity timeout handler” state, in the case where either:

* + the physical video monitor is currently turned off;
  + the “user input inactivity timeout” policy specifies “Start a password protected screen saver”;
  + the screen saver grace period is non-zero;
  + the screen saver grace period has elapsed;
  + the default screen saver application process started by the service has not terminated;
  + the “user input inactivity timeout” policy specifies “Locking of the display areas”;
  + the screen saver grace period is non-zero;
  + the screen saver grace period has elapsed;
  + the default screen saver application process started by the service has not terminated.

The instance of the window logon state maintaining service conducts the following:

* generates the [Event ID 4803](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_SCREENSAVER\_DISMISSED) “The screen saver was dismissed” audit record in the hard audit store;
* initiates a locking of the display areas;
* switches to the secure display area,

before exiting from the “Post logged on inactivity timeout handler” state, in the case where either:

* + the “user input inactivity timeout” policy specifies “Start a password protected screen saver”;
  + the administrator specified screen saver application process started by the service has terminated after the screen saver grace period has elapsed;
  + the “user input inactivity timeout” policy specifies “Start the screen saver”;
  + the “screen saver” policy specifies “the screen saver application process runs on its own screen saver display area”;
  + the administrator specified screen saver application process started by the service has terminated after the screen saver grace period has elapsed.

The instance of the window logon state maintaining service conducts the following:

* generates the [Event ID 4803](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_SCREENSAVER\_DISMISSED) “The screen saver was dismissed” audit record in the hard audit store;
* initiates a locking of the display areas;
* switches back to the user interactive application display area,

before exiting from the “Post logged on inactivity timeout handler” state, in the case where either:

* + the “user input inactivity timeout” policy specifies “Start a password protected screen saver”;
  + the administrator specified screen saver application process started by the service has terminated before the screen saver grace period has elapsed;
  + the “user input inactivity timeout” policy specifies “Start the screen saver”;
  + the “screen saver” policy specifies “the screen saver application process runs on its own screen saver display area”;
  + the administrator specified screen saver application process started by the service has terminated before the screen saver grace period has elapsed.

The instance of the window logon state maintaining service conducts the following:

* generates the [Event ID 4803](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_SCREENSAVER\_DISMISSED) “The screen saver was dismissed” audit record in the hard audit store,

before exiting from the “Post logged on inactivity timeout handler” state, in the case where

* the “user input inactivity timeout” policy specifies “Start the screen saver”;
* the “screen saver” policy specifies “the screen saver application process runs on the user interactive application display area”.

The instance of the window logon state maintaining service conducts the following:

* initiates a forced logoff for the already authenticated interactive user,

before exiting from the “Post logged on inactivity timeout handler” state, in the case where

* the “user input inactivity timeout” policy specifies “Log off”;
* the screen saver grace period is non-zero;
* the screen saver grace period has elapsed;
* the default screen saver application process started by the service has not terminated.

The default screen saver application process started by the instance of the window logon state maintaining service terminates itself upon receiving the following window messages from the local Windows OS window manager:

* a mouse button is physically pressed ([WM\_LBUTTONDOWN](http://msdn.microsoft.com/en-us/library/ms645607.aspx), [WM\_MBUTTONDOWN](http://msdn.microsoft.com/en-us/library/ms645610(VS.85).aspx), [WM\_RBUTTONDOWN](http://msdn.microsoft.com/en-us/library/ms646242(VS.85).aspx));
* the mouse is physically moved by a specific threshold distance ([WM\_MOUSEMOVE](http://msdn.microsoft.com/en-us/library/ms645616(VS.85).aspx));
* a non-system keyboard key is pressed, where a non-system key is a keyboard key that is pressed when the ALT key is not pressed ([WM\_KEYDOWN](http://msdn.microsoft.com/en-us/library/ms646280(VS.85).aspx));
* the translated character of the non-system key being pressed ([WM\_CHAR](http://msdn.microsoft.com/en-us/library/ms646276(VS.85).aspx)).

The administrator specified screen saver application process started by the instance of the window logon state maintaining service terminates itself upon receiving the following window messages from the local Windows OS window manager:

* a mouse button is physically pressed ([WM\_LBUTTONDOWN](http://msdn.microsoft.com/en-us/library/ms645607.aspx), [WM\_MBUTTONDOWN](http://msdn.microsoft.com/en-us/library/ms645610(VS.85).aspx), [WM\_RBUTTONDOWN](http://msdn.microsoft.com/en-us/library/ms646242(VS.85).aspx));
* the mouse is physically moved by a specific threshold distance ([WM\_MOUSEMOVE](http://msdn.microsoft.com/en-us/library/ms645616(VS.85).aspx));
* a non-system keyboard key is pressed, where a non-system key is a keyboard key that is pressed when the ALT key is not pressed ([WM\_KEYDOWN](http://msdn.microsoft.com/en-us/library/ms646280(VS.85).aspx));
* the translated character of the non-system key being pressed ([WM\_CHAR](http://msdn.microsoft.com/en-us/library/ms646276(VS.85).aspx)).

The instance of the window logon state maintaining service conducts the following:

* initiates a locking of the display areas;
* switches to the secure display area,

before exiting from the “Post logged on inactivity timeout handler” state, in the case where either:

* + the physical video monitor is currently turned off;
  + the “user input inactivity timeout” policy specifies “Start a password protected screen saver”;
  + the screen saver grace period is zero;
  + the “user input inactivity timeout” policy specifies “Locking of the display areas”;
  + the screen saver grace period is zero;
  + the “user input inactivity timeout” policy is not set to any valid option.

The instance of the window logon state maintaining service conducts the following:

* initiates a forced logoff for the already authenticated interactive user,

before exiting from the “Post logged on inactivity timeout handler” state, in the case where

* the “user input inactivity timeout” policy specifies “Log off”;
* the screen saver grace period is zero.

### Transitions from the “Post logged on inactivity timeout handler” state

The “Post logged on inactivity timeout handler” state supports the following state transitions.

* If the “Post logged on inactivity timeout handler” state transitions to the “User logged on” state so that the already authenticated interactive user has his/her interactive application display area visible and ready to receive user input again, then either:
  + the service must have switched back to the user interactive application display area;
  + the “user input inactivity timeout” policy must have said “Nothing”;
  + the “user input inactivity timeout” policy must have said “Log off”;
  + the “user input inactivity timeout” policy must have said “Start the screen saver”, and the “screen saver” policy must have said “the screen saver application process runs on the user interactive application display area”, and the screen saver application process must have terminated.
* If the “Post logged on inactivity timeout handler” state transitions to the “User logged on” state and then to the “Initiate lock on display areas” state, then
  + the service must have switched to the secure display area.
* If the “Post logged on inactivity timeout handler” state transitions to the “User logged on” state and then to the “Welcome” state[[25]](#footnote-26), then either:
  + the local shutdown initiation service (as part of ininit.exe) must have notified a logoff or shutdown;
  + a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service.
* If the “Post logged on inactivity timeout handler” state transitions to the “User logged on” state and then to the “Session disconnected” state[[26]](#footnote-27), then
  + the Local Window Terminal Service (LWTS) must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in;
  + the service must have switched back to the user interactive application display area.
* If the “Post logged on inactivity timeout handler” state transitions to the “User logged on” state and then to the “Post logged on hibernate” state[[27]](#footnote-28), then
  + there must have been a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service (aka LogonUI.exe) to display the below messages:
    - “Windows is shutting down...” corresponding to data saving;
    - “Undocking and going to sleep...”;
    - “Hibernating...”
    - “Going to sleep...” corresponding to the entering of the “standing by” Windows OS system power state.
* If the “Post logged on inactivity timeout handler” state transitions to the “Trusted path activated” state, then
  + there must have been either:
    - the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
    - the notification of a trusted path simulation from an authorized subject.

### “Post logged on resume” state

The “Post logged on resume” state is transitioned from the “Post logged on hibernate” state after

* the arrival of a notification of the user input inactivity timeout which has been mediated by the local Windows OS window manager of the same window terminal session;
* either:
  + the “user input inactivity timeout” policy specifies “Start the screen saver” and the “screen saver” policy specifies “the screen saver application process runs on the user interactive application display area”;
  + the “user input inactivity timeout” policy specifies “Nothing”.

The instance of the window logon state maintaining service reacts to one of the following notifications, upon entering the “Post logged on resume” state.

* A notification originated from the Windows OS power manager or another authorized source to abort the entering of the shutdown, hibernation, or standing-by power state.
* A notification originated from the Windows OS power manager or another authorized source to resume from the hibernation or the standing-by power state and to request the window logon user interface service (aka LogonUI.exe) to display the below messages.
  + “Windows is resuming...” from the last hibernation.
  + “Windows is resuming...” from the last “standing by” system power state.

While still remaining in the “Post logged on resume” state, if the instance of the window logon state maintaining service requests the window logon user interface service (aka LogonUI.exe) to display resume messages on the secure display area, then

* there must have been a notification originated from the Windows OS power manager or another authorized source to resume from the hibernation or the standing-by power state and to request the window logon user interface service (aka LogonUI.exe) to display the below messages.
  + “Windows is resuming...” from the last hibernation.
  + “Windows is resuming...” from the last “standing by” system power state.

### Transitions from the “Post logged on resume” state

The “Post logged on resume” state supports the following state transitions.

* If the “Post logged on resume” state transitions to the “User logged on” state, then
  + there must have been a notification originated from the Windows OS power manager or another authorized source to abort the entering of the shutdown, hibernation, or standing-by power state.

At the entry of the “User logged on” state, following the transition from the “Post logged on resume” state, the current active display area may be the secure display area or the user interactive application display area.

The following state transition path results in the secure display area at the entry of the “User logged on” state.



The following state transition path results in the user interactive application display area at the entry of the “User logged on” state.



### “Initiate lock during resume” state

If the “Initiate lock during resume” state is transitioned from the “Post logged on hibernate” state then either:

* there must have been a notification of an authorized subject’s initiation of a locking of the display areas or there must have been a request originated from an authorized subject to lock the display areas;
* there must have been a notification of the user input inactivity timeout which has been mediated by the local Windows OS window manager of the same window terminal session, and either
  + the “user input inactivity timeout” policy specifies “Start the screen saver” and the “screen saver” policy specifies “the screen saver application process runs on its own screen saver display area”;
  + the “user input inactivity timeout” policy specifies “Start a password protected screen saver”;
  + the “user input inactivity timeout” policy specifies “Locking of the display areas”;
  + the “user input inactivity timeout” policy specifies “Log off”.

The instance of the window logon state maintaining service conducts the following:

* switches to the user interactive application display area,

upon entering the “Initiate lock during resume” state, in the case where

* the administrator specified “[display area locking](http://technet2.microsoft.com/windowsserver/en/library/c7410c7f-d6cf-400b-9604-f1e2b4fe1d2c1033.mspx?mfr=true)” policy is disabled.

The instance of the window logon state maintaining service conducts the following:

* switches to the secure display area;
* generates the [Event ID 4800](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_WORKSTATION\_LOCKED) “The (workstation) display areas are locked” audit record in the hard audit store;

upon entering the “Initiate lock during resume” state, in the case where

* the administrator specified “[display area locking](http://technet2.microsoft.com/windowsserver/en/library/c7410c7f-d6cf-400b-9604-f1e2b4fe1d2c1033.mspx?mfr=true)” policy is enabled.

### Transitions from the “Initiate lock during resume” state

The “Initiate lock during resume” state supports the following state transitions.

* If the “Initiate lock during resume” state transitions to the “Post logged on resume” state, then
  + the instance of the window logon state maintaining service must have switched to the user interactive application display area.
* If the “Initiate lock during resume” state transitions to the “Locked during resume” state, then
  + the instance of the window logon state maintaining service must have switched to the secure display area.

### “Locked during resume” state

The instance of the window logon state maintaining service reacts to one of the following notifications, upon entering the “Locked during resume” state.

* A notification originated from the Windows OS power manager or another authorized source to abort the entering of the shutdown, hibernation, or standing-by power state.
* A notification originated from the Windows OS power manager or another authorized source to resume from the hibernation or the standing-by power state and to request the window logon user interface service (aka LogonUI.exe) to display the below messages.
  + “Windows is resuming...” from the last hibernation.
  + “Windows is resuming...” from the last “standing by” system power state.

While still remaining in the “Locked during resume” state, if the instance of the window logon state maintaining service requests the window logon user interface service (aka LogonUI.exe) to display resume messages on the secure display area, then

* there must have been a notification originated from the Windows OS power manager or another authorized source to resume from the hibernation or the standing-by power state and to request the window logon user interface service (aka LogonUI.exe) to display the below messages.
  + “Windows is resuming...” from the last hibernation.
  + “Windows is resuming...” from the last “standing by” system power state.

### Transitions from the “Locked during resume” state

The “Locked during resume” state supports the following state transitions.

* If the “Locked during resume” state transitions to the “Locked” state, then
  + there must have been a notification originated from the Windows OS power manager or another authorized source to abort the entering of the shutdown, hibernation, or standing-by power state.

### “Inactivity timeout handler during locked” state

The behaviors of this state have been described earlier in the ““Inactivity timeout handler during locked” state” section as a part of the justification text for addressing the Commercial Grade OS Requirement Set “3.2.1.2” requirement.

### Human interface device input and video output separation maintained by the Windows OS window manager(s)

The Windows OS window manager (win32k.sys) of every window terminal session has the following two local interfaces for being informed about its window terminal session connection and disconnection respectively by only the [Windows OS Client Server Runtime Subsystem (csrss.exe)](http://msdn.microsoft.com/en-us/library/ms818751.aspx) residing in the same window terminal session.

* “Remote Reconnect” function;
* “Remote Disconnect” function.

When the Local Window Terminal Service (LWTS) (lsm.exe) initiates the (re)connection of a window terminal session (say Session n), the Session n Windows OS window manager is informed about the (re)connection via the [Windows OS Client Server Runtime Subsystem (csrss.exe)](http://msdn.microsoft.com/en-us/library/ms818751.aspx) residing in Session n. The Session n Windows OS window manager registers with the Windows OS plug and play manager for notification of the events:

* “Event Category Target Device Change”;
* “Event Category Device Interface Change”,

being detected for local mice, keyboards and other human interface devices (HIDs). When a registered device is ready for reading, the Session n Windows OS window manager starts reading from the device via its internal “Start Device Read” function. For mice, there is the “Process Mouse Input” internal read function. For keyboards, there is the “Process Keyboard Input” internal read function. For HIDs, there is the “Process Hid Input” internal read function. Mouse events and keyboard events arriving through these local devices are not marked as injected. Raw HID events are delivered through [WM\_INPUT](http://msdn.microsoft.com/en-us/library/ms645590(VS.85).aspx) messages to the windows whose process owners have successfully been registered for receiving raw HID events via [RegisterRawInputDevices()](http://msdn.microsoft.com/en-us/library/ms645600(VS.85).aspx).

When the Local Window Terminal Service (LWTS) (lsm.exe) initiates the disconnection of Session n, the Session n Windows OS window manager detaches all local input devices from Session n. Therefore, the Session n Windows OS window manager would not receive any input data generated by local hardware devices such as keyboards, mice, and other HIDs. Consequently, there is no delivery of input data to the Session n Windows OS window manager’s consumer subjects.

As a result, the Session n Windows OS window manager mediates input data generated by local hardware devices such as keyboards, mice, and other HIDs before delivering them to the Session n Windows OS window manager’s consumer subjects as authentic user input. Furthermore, when Session n is disconnected, input data generated by local hardware devices such as keyboards, mice, and other HIDs are not delivered to the Session n Windows OS window manager’s consumer subjects.

The Session n Windows OS window manager updates its “timeLastInputMessage” variable with the current time in the “Process Mouse Input” internal read function for non injected events, in the “Process Keyboard Input” internal read function for non injected events, in the “Process Hid Input” internal read function, and in the processing of the “Power State Resume Apps” notification from the Windows OS power manager. Therefore, the Session n Windows OS window manager always maintains the time of last input message from local hardware devices such as keyboards, mice, and other HIDs. The current value of the time of last input message from the “timeLastInputMessage” variable is then used by the Session n Windows OS window manager to determine if it should notify the instance of the window logon state maintaining service residing in Session n about the user input inactivity timeout.

Carrying out drawing operations on a video display device requires a [device context handle (HDC)](http://msdn.microsoft.com/en-us/library/ms533227(VS.85).aspx). A device context handle is only created based on the monitor device that is updated by the Session n Windows OS window manager whenever it processes a session disconnection and a session reconnection request from the [Windows OS Client Server Runtime Subsystem (csrss.exe)](http://msdn.microsoft.com/en-us/library/ms818751.aspx). During the disconnection of Session n, the monitor device is associated with the disconnect display device (DISPLAY\_DEVICE\_DISCONNECT). During the reconnection of Session n, the monitor device is associated with a local attached display device. As a result, when Session n is connected, the Session n Windows OS window manager mediates graphical display output instruction data created by the Session n Windows OS window manager’s consumer subjects before sending them to the attached local display devices. Similarly, when Session n is disconnected, the Session n Windows OS window manager mediates graphical display output instruction data created by the Session n Windows OS window manager’s consumer subjects before sending them to the disconnect display device (DISPLAY\_DEVICE\_DISCONNECT), which does nothing with the graphical display output instruction data.

### “Screen saver timeout” policy

The Windows OS supports the “screen saver timeout” policy, which specifies how much user input idle time must elapse before the Windows OS window manager of a specific window terminal session notifies the instance of the window logon state maintaining service of the same session about the user input inactivity timeout. The default “screen saver timeout” value is 600 seconds from the default user profile (HKEY\_USERS\.Default\Control Panel\Desktop).

By default, only an administrator is allowed to configure the “screen saver timeout” policy.

### User input inactivity timeout notification from the Windows OS window manager

The Session n Windows OS window manager maintains a timer with a timer function, which is executed repeatedly in a “1000 millisecond” period cycle. The timer function compares the difference between the current time and the time of last input message from local hardware devices such as keyboards, mice, and other HIDs (i.e. the current saved value in the “timeLastInputMessage” variable). If the difference is bigger than the “screen saver timeout” value of the “screen saver timeout” policy, then the Windows OS window manager of a specific window terminal session notifies the instance of the window logon state maintaining service of the same session about the user input inactivity timeout.

### Summary of the Windows OS justification for meeting “3.2.1.5”

As the “screen saver timeout” value of the “screen saver timeout” policy is the authorized administrator specified time interval of user inactivity, the above have explained the transitions to the following states after the authorized administrator specified time interval of user inactivity expires:

* the “Locked” state;
* the “Initiate lock on display areas” state, which also transitions to the “Locked” state as explained in the “Transitions from the “Initiate lock on display areas” state” section.

Due to the switching to the secure display area before entering the “Locked” state, the user interactive application display area, belonging to the instance of the window logon state maintaining service, is not visible and is not receiving user input.

Therefore, the Windows OS accomplishes the following actions of this “3.2.1.5” requirement.

1. Clearing or overwriting display devices, making the current contents unreadable;
2. Disabling any activity from the user’s data access/display devices other than unlocking the interactive session.

## Addressing 3.2.1.6 “The OS shall provide the ability to deny interactive session establishment based on time and day”

This requirement is addressed by the Windows OS as follows.

The possible action values which could be used to determine the action (which is the action to be taken by the instance of the window logon state maintaining service when the user’s allowable logon hours expire) due to the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy are:

* “nothing”;
* lock the display areas;
* disconnect the window terminal session where the instance of the window logon state maintaining service resides in;
* forced logoff for the already authenticated interactive user.

As explained in the ““Report unlock failure” state” section, the window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the user account is not allowed to logon at the current time due to the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy assigned to the account by an administrator ([STATUS\_INVALID\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/cc704588.aspx)).

* “Your account has time restrictions that prevent you from logging on at this time.”

Also, the “Report unlock failure” state transitions to the “Locked” state after displaying the above message according to the “Transitions from the “Report unlock failure” state” section.

As explained in the ““Notify logon subscribers” state” section, the instance of the window logon state maintaining service conducts the following, upon entering the “Notify logon subscribers” state.

* The enforcement of the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy for the already authenticated interactive user who is not an administrator.

Furthermore, as explained in the ““User logged on” state” section, while still remaining in the “User logged on” state, if the instance of the window logon state maintaining service conducts the following:

* initiates the locking of the display areas for the already authenticated interactive user who is not an administrator, in the case where the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy action value is “lock the display areas”;
* requests the Local Window Terminal Service (LWTS) to disconnect the window terminal session where the instance of the window logon state maintaining service resides in for the already authenticated interactive user who is not an administrator, in the case where the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy action value is “disconnect the window terminal session where the instance of the window logon state maintaining service resides in”;
* initiates a forced logoff for the already authenticated interactive user who is not an administrator, in the case where the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy action value is “forced logoff”,

then

* the already authenticated user’s allowable logon hours must have expired in the case where the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy is not “nothing”.

As a result, this Commercial Grade OS Requirement Set “3.2.1.6” requirement is met.

## Addressing 3.2.1.7 “Upon successful interactive session establishment, the OS shall display to the authorized user a specific set of user access history information elements”

The Commercial Grade OS Requirement Set requires the following specific set of user access history information elements to be displayed to the authorized user:

1. Date, time, and location of that user’s last successful interactive session establishment;
2. The number of unsuccessful (interactive session establishment) authentication attempts for that user since last successful interactive session establishment.

This requirement is addressed by the Windows OS as follows.

As explained in the ““Report last logon to user” state” section, the window logon user interface service displays one of the following message box messages on the secure display area to collect the interactive user’s response after the success authentication of the interactive user for this current interactive logon attempt, in the case where the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy is enabled.

* “<user account name> This is the first time you have interactively logged on to this account.”
* “<user account name> This is the first time you have interactively logged on to this account. Unsuccessful Logon: The last unsuccessful interactive logon attempt on this account was: <Date/Time of the previous failed logon>. The number of unsuccessful interactive logon attempts since your account was created: <failed attempt count>.”
* “<user account name> Successful Logon: The last time you interactively logged on to this account was: <Date/Time of the previous success logon>. Unsuccessful Logon: There have been no unsuccessful interactive logon attempts with this account since your last interactive logon.”
* “<user account name> Successful Logon: The last time you interactively logged on to this account was: <Date/Time of the previous success logon>. Unsuccessful Logon: The last unsuccessful interactive logon attempt on this account was: <Date/Time of the previous failed logon>. The number of unsuccessful interactive logon attempts since your last interactive logon: <failed attempt count>.”
* “Security policies on this computer are set to display information about the last interactive logon. However, this information could not be retrieved. ”

In the message box containing any of the above messages about the information of previous logons, the window logon user interface service provides only the “OK” button for collecting the interactive user’s response.

The window logon user interface service keeps any of the above messages about the information of previous logons as being displayed on the secure display area until the interactive user selects one of the following as his/her response, or until the dialog timeout occurs.

* “OK”,

as presented on the secure display area.

As explained in the ““Report last logon to user after unlock” state” section, the window logon user interface service displays one of the following message box messages on the secure display area to collect the interactive user’s response after the success authentication of the interactive user for this current unlock attempt, in the case where the “[Windows Logon Options: display information about previous logons during user logon](http://technet2.microsoft.com/windowsvista/en/library/770b1181-be7e-4ef4-809b-e74ecc52c2d41033.mspx?mfr=true)” policy is enabled.

* “<user account name> Successful Logon: The last time you interactively logged on to this account was: <Date/Time of the previous success logon>. Unsuccessful Logon: There have been no unsuccessful interactive logon attempts with this account since your last interactive logon.”
* “<user account name> Successful Logon: The last time you interactively logged on to this account was: <Date/Time of the previous success logon>. Unsuccessful Logon: The last unsuccessful interactive logon attempt on this account was: <Date/Time of the previous failed logon>. The number of unsuccessful interactive logon attempts since your last interactive logon: <failed attempt count>.”
* “Security policies on this computer are set to display information about the last interactive logon. However, this information could not be retrieved. ”

In the message box containing any of the above messages about the information of previous logons, the window logon user interface service provides only the “OK” button for collecting the interactive user’s response.

The window logon user interface service keeps any of the above messages about the information of previous logons as being displayed on the secure display area until the interactive user selects one of the following as his/her response, or until the dialog timeout occurs.

* “OK”,

as presented on the secure display area.

The above shows that this Commercial Grade OS Requirement Set “3.2.1.7” requirement is met except for the fact that the above “Report last logon to user” or “Report last logon to user after unlock” message box messages do not include the location information of the last time the user interactively logged on to the user account successfully.

Due to the potential complexity of the distributed environments where the Windows OS machines are deployed in, IP address is the only viable, but still imperfect[[28]](#footnote-29), option for representing a user/client logging on location. Indeed, as explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.3.5” requirement, the Kerberos KDC residing on a Windows OS domain controller attempts to capture the client IP address in its audit generation of the [Event ID 4768](http://support.microsoft.com/kb/947226) “A Kerberos authentication ticket (TGT) was requested” (SE\_AUDITID\_AS\_TICKET/SE\_AUDITID\_ETW\_AS\_TICKET). However, the IP address information in the audit records (when available) is really meant for the benefit of administrators when they need to conduct security related investigations.

Furthermore, given that IPV6 network infrastructure is relatively common nowadays, the IPV6 address format would be the common case for the IP address information. An IP V6 address has the form as something like “2001:4898:0:fff:200:5efe:157.59.22.29”. Therefore, it seems to be an extra burden for a standard user to discover and then remember the IPV6 address assigned to the Windows OS machine where he/she is conducting the logging on. Additionally, if the standard user is using a wireless network and/or a virtual private network (VPN), then his/her Windows OS machine is assigned other IPV6 addresses for the respective purposes. It is therefore necessary that the user is taught on how to discover which relevant IPV6 address to remember.

It is believed that if the standard user is attentive enough to discover and remember his/her last successful logon Windows OS machine IP address, he/she would be attentive enough to remember the exact date/time of his/her last successful logon. Given the exact date/time of his/her last successful logon that the standard user can remember for comparing with the date/time of his/her last successful logon as shown in the above “Report last logon to user” or “Report last logon to user after unlock” message box messages, it seems that the usefulness of his/her last successful logon Windows OS machine IP address in the same message box messages is fairly marginal.

As a result, we believe that the lack of the last successful logon Windows OS machine IP address information in the above “Report last logon to user” or “Report last logon to user after unlock” message box messages does not reduce the effectiveness of the intents behind this Commercial Grade OS Requirement Set “3.2.1.7” requirement.

## Addressing 3.2.1.8 “The OS shall not erase the user access history information elements from the authorized user interface without giving the authorized user the opportunity to review the information elements”

This requirement is addressed by the Windows OS as follows.

As explained in the ““Report last logon to user” state” section and the ““Report last logon to user after unlock” state” section, the window logon user interface service keeps any of the above “Report last logon to user” or “Report last logon to user after unlock” message box messages about the information of previous logons as being displayed on the secure display area until the interactive user selects one of the following as his/her response, or until the dialog timeout occurs.

* “OK”,

as presented on the secure display area.

As explained in the “Transitions from the “Report last logon to user” state” section and the “Transitions from the “Report last logon to user after unlock” state” section, the interactive user must select the “OK” button for completing the remaining of his/her logon and his/her display area unlocking procedures respectively.

Consequently, this Commercial Grade OS Requirement Set “3.2.1.8” requirement is met.

# Meeting the “Identification and Authentication User Interface Security Management Requirements”

In the Commercial Grade OS Requirement Set, there are 5 individual management requirements under the heading of “Identification and Authentication User Interface Security Management Requirements”. They are listed as “3.2.2.n”, where n = 1, 2, 3, 4, and 5.

## Addressing 3.2.2.1 “The OS shall provide authorized administrators with the ability to specify an advisory notice”

This requirement is addressed by the Windows OS as follows.

If a subject requests to modify the legal notices which would be displayed in the “legal notices” message box message, as explained in the ““Display legal notices” state” section of this paper, on the secure display area as a Windows OS behavior item, then the behavior item must be allowed by the administrator specified policy. The following registry keys grant only an administrator the KEY\_SET\_VALUE right.

* HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System \LegalNoticeCaption;
* HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows\CurrentVersion\Policies\System \LegalNoticeText;
* HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\LegalNoticeCaption;
* HKEY\_LOCAL\_MACHINE\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\LegalNoticeText.

Consequently, this Commercial Grade OS Requirement Set “3.2.2.1” requirement is met.

## Addressing 3.2.2.2 “The OS shall provide authorized administrators with the ability to terminate an interactive session”

This requirement is addressed by the Windows OS as follows.

### Window terminal session specific “Window terminal session access” policy

The “Window terminal session access” policy of a specific local or remote window terminal session consists of the following attributes.

* Any one of the following access permissions may be granted or denied to an authenticated user:
  + Permission to query information about a window terminal session ([WINSTATION\_QUERY](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
  + Permission to reset or end the network listener associated with a window terminal session, or to logoff a window terminal session ([WINSTATION\_RESET](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
  + Permission to create and use a virtual channel associated with a window terminal session ([WINSTATION\_VIRTUAL](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
  + Permission to shadow or remotely control a window terminal session which belongs to another user account ([WINSTATION\_SHADOW](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
  + Permission to logon to a window terminal session ([WINSTATION\_LOGON](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
  + Permission to pop a message box on the current display area of a window terminal session ([WINSTATION\_MSG](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx))
  + Permission to connect to another session ([WINSTATION\_CONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
  + Permission to disconnect a session ([WINSTATION\_DISCONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
* Audit the granting or denial of any one of the following access permissions to an authenticated user:
  + Permission to query information about a window terminal session ([WINSTATION\_QUERY](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
  + Permission to reset or end the network listener associated with a window terminal session, or to logoff a window terminal session ([WINSTATION\_RESET](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
  + Permission to create and use a virtual channel associated with a window terminal session ([WINSTATION\_VIRTUAL](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
  + Permission to shadow or remotely control a window terminal session which belongs to another user account ([WINSTATION\_SHADOW](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
  + Permission to logon to a window terminal session ([WINSTATION\_LOGON](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
  + Permission to pop a message box on the current display area of a window terminal session ([WINSTATION\_MSG](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)); Permission to connect to another session ([WINSTATION\_CONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx));
  + Permission to disconnect a session ([WINSTATION\_DISCONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)).

By default, an administrator possesses all the above access permissions of the “Window terminal session access” policy for any local or remote window terminal session.

If a subject requests to define the default “Window Terminal Session Access” policy for a local or remote window terminal session,

as a behavior item, then

• the behavior item must be allowed by the administrator specified policy.

The following registry key value defining the behavior does not allow the non privileged user to write.

• HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\DefaultSecurity.

### Capabilities available administrators

As explained in the section of “Summary of the Windows OS justification for meeting the requirement for an administrator to unlock and terminate a user session”, the following dialogs are available to an administrator during the “Get session confirmation from LWTS” state.

* the “end another session” dialog;
* the “disconnect another session” dialog;
* the “select one of the sessions to force disconnection” dialog;
* the “select one of the sessions to request disconnection from the selected session’s logged on user” dialog.

Furthermore, as described in the [Microsoft publication: “Security Functional Assertions of the “User Interaction based on Windowing” Scenario of a Modern Operating System”](http://download.microsoft.com/download/e/d/b/edbb17fb-580d-49a4-b66c-8726cf446a86/User%20Interaction.doc), the Local Window Terminal Service (LWTS) (lsm.exe) and the Windows OS remote window terminal service (termsrv.dll) mediate the behavior(s) for:

* remote and local window terminal session disconnection;
* remote and local window terminal session reconnection from a (distinct) source window terminal session.

### Remote or local window terminal session disconnection

If a subject requests to disconnect a specific connected window terminal session (say Session T), where

* the subject is granted the disconnection right ([WINSTATION\_DISCONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to the specified Session T according to Session T’s “Window terminal session access” policy;
* the specified Session T is
  + a remote window terminal session,

then

* the Windows OS remote window terminal service requests Session T’s Windows OS window manager to disconnect Session T from the established remote network stack so that no contents (i.e. window graphical interfaces) from the display areas of Session T are displayed to the remote interactive user and no input from the remote interactive user is accepted;
* the Windows OS remote window terminal service notifies the remote network RDP client about the termination of the client’s RDP connection to Session T;
* the network connection stack established for communicating with the remote network RDP client is closed;
* the Windows OS remote window terminal service generates the [Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value) audit record which identifies Session T after the Session T disconnection occurs.

If a subject requests to disconnect a specific connected window terminal session (say Session T), where

* the subject is granted the disconnection right (WINSTATION\_DISCONNECT) to the specified Session T according to Session T’s “Window terminal session access” policy;
* the specified Session T is
  + a local window terminal session,

then

* the Local Window Terminal Service (LWTS) requests Session T’s Windows OS window manager to disconnect Session T from the local display terminal, and connect Session T to the disconnect display device (DISPLAY\_DEVICE\_DISCONNECT), so that no contents (i.e. window graphical interfaces) from the display areas of Session T are displayed to the local interactive user and no input from the local interactive user is accepted;
* the Local Window Terminal Service (LWTS) generates the [Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value) audit record which identifies Session T after the Session T disconnection occurs;
* a new local window terminal session is created, where
  + the window manager (win32k.sys) instance of this new window terminal session starts;
  + the window logon state maintaining service (winlogon.exe) instance of this new window terminal session starts;
  + the window logon user interface service (LogonUI.exe) instance of this new window terminal session starts;
  + this new window terminal session is connected to the local display terminal so that the “Welcome” message from its window logon user interface service instance is displayed to the local interactive user in the “Welcome” state of its window logon state maintaining service instance.

By default, an administrator possesses the disconnection right ([WINSTATION\_DISCONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to any local or remote window terminal session.

### Remote or local window terminal session reconnection from a (distinct) source window terminal session

If a subject requests to

* disconnect a specific remote connected target window terminal session (say Session T) from its current network stack (say Stack T);
* disconnect a specific remote connected source window terminal session (say Session S) from its current network stack (say Stack S);
* connect Session S to Stack T,

where

* either:
  + the subject is the remote interactive user logged on to Session S;
  + the subject possesses the valid authentication data (e.g. password) of the remote interactive user already logged on to Session S;
* the subject is granted the connection right ([WINSTATION\_CONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to the specified Session S according to Session S’s “Window terminal session access” policy;
* the subject is granted the disconnection right ([WINSTATION\_DISCONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to the specified Session T according to Session T’s “Window terminal session access” policy,

then

* the Windows OS remote window terminal service requests Session S’s Windows OS window manager to disconnect Session S from Stack S so that no contents (i.e. window graphical interfaces) from the display areas of Session S are displayed to the remote interactive user of Session S and no input from the remote interactive user of Session S is accepted;
* the Windows OS remote window terminal service notifies the remote network RDP client about the termination of the client’s RDP connection to Session S;
* network Stack S for communicating with the remote network RDP client is closed;
* the Windows OS remote window terminal service generates the [Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value) audit record which identifies Session S after the Session S disconnection occurs;
* the Windows OS remote window terminal service requests Session T’s Windows OS window manager to disconnect Session T from Stack T so that no contents (i.e. window graphical interfaces) from the display areas of Session T are displayed to the remote interactive user of Session T and no input from the remote interactive user of Session T is accepted;
* the Windows OS remote window terminal service generates the [Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value) audit record which identifies Session T after the Session T disconnection occurs;
* the Windows OS remote window terminal service requests Session S’s Windows OS window manager to connect Session S to Stack T so that contents (i.e. window graphical interfaces) from the display areas of Session S are displayed to the remote interactive user and input from the remote interactive user is accepted;
* the Windows OS remote window terminal service generates the [[Event ID 4778](http://support.microsoft.com/kb/947226) “A session was reconnected to a Window Station” (SE\_AUDITID\_ETW\_SESSION\_RECONNECTED\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record which identifies Session S after the Session S reconnection occurs;
* Session T becomes a disconnected window terminal session.

If a subject requests to

* disconnect a specific remote connected target window terminal session (say Session T) from its current network stack (say Stack T);
* connect a specific disconnected source window terminal session (say Session S) to Stack T,

where

* either:
  + the subject is the remote interactive user logged on to Session S;
  + the subject possesses the valid authentication data (e.g. password) of the remote interactive user already logged on to Session S;
* the subject is granted the connection right ([WINSTATION\_CONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to the specified Session S according to Session S’s “Window terminal session access” policy;
* the subject is granted the disconnection right ([WINSTATION\_DISCONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to the specified Session T according to Session T’s “Window terminal session access” policy,

then

* the Windows OS remote window terminal service requests Session T’s Windows OS window manager to disconnect Session T from Stack T so that no contents (i.e. window graphical interfaces) from the display areas of Session T are displayed to the remote interactive user of Session T and no input from the remote interactive user of Session T is accepted;
* the Windows OS remote window terminal service generates the [Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value) audit record which identifies Session T after the Session T disconnection occurs;
* the Windows OS remote window terminal service requests Session S’s Windows OS window manager to connect Session S to Stack T so that contents (i.e. window graphical interfaces) from the display areas of Session S are displayed to the remote interactive user (who previously was seeing window graphical interfaces from Session T) and input from the remote interactive user is accepted;
* the Windows OS remote window terminal service generates the [[Event ID 4778](http://support.microsoft.com/kb/947226) “A session was reconnected to a Window Station” (SE\_AUDITID\_ETW\_SESSION\_RECONNECTED\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record which identifies Session S after the Session S reconnection occurs;
* Session S becomes a remote connected window terminal session;
* Session T becomes a disconnected window terminal session.

If a subject requests to

* disconnect a specific local connected target window terminal session (say Session T) from the local display terminal;
* disconnect a specific remote connected source window terminal session (say Session S) from its current network stack (say Stack S);
* connect Session S to the local display terminal,

where

* either:
  + the subject is the remote interactive user logged on to Session S;
  + the subject possesses the valid authentication data (e.g. password) of the remote interactive user already logged on to Session S;
* the subject is granted the connection right ([WINSTATION\_CONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to the specified Session S;
* the subject is granted the disconnection right ([WINSTATION\_DISCONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to the specified Session T,

then

* the Windows OS remote window terminal service requests Session S’s Windows OS window manager to disconnect Session S from Stack S so that no contents (i.e. window graphical interfaces) from the display areas of Session S are displayed to the remote interactive user of Session S and no input from the remote interactive user of Session S is accepted;
* the Windows OS remote window terminal service notifies the remote network RDP client about the termination of the client’s RDP connection to Session S;
* network Stack S for communicating with the remote network RDP client is closed;
* the Windows OS remote window terminal service generates the [Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value) audit record which identifies Session S after the Session S disconnection occurs;
* the Windows OS remote window terminal service requests Session T’s Windows OS window manager to disconnect Session T from the local display terminal so that no contents (i.e. window graphical interfaces) from the display areas of Session T are displayed to the local interactive user of Session T and no input from the local interactive user of Session T is accepted;
* the Windows OS remote window terminal service generates the [Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value) audit record which identifies Session T after the Session T disconnection occurs;
* the Windows OS remote window terminal service requests Session S’s Windows OS window manager to connect Session S to the local display terminal so that contents (i.e. window graphical interfaces) from the display areas of Session S are displayed to the local interactive user (who previously was seeing window graphical interfaces from Session T) and input from the local interactive user is accepted;
* the Windows OS remote window terminal service generates the [[Event ID 4778](http://support.microsoft.com/kb/947226) “A session was reconnected to a Window Station” (SE\_AUDITID\_ETW\_SESSION\_RECONNECTED\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record which identifies Session S after the Session S reconnection occurs;
* Session S becomes a local connected window terminal session;
* Session T becomes a disconnected window terminal session.

If a subject requests to

* disconnect a specific local connected target window terminal session (say Session T) from the local display terminal;
* connect a specific disconnected source window terminal session (say Session S) to the local display terminal,

where

* either:
  + the subject is the remote interactive user logged on to Session S;
  + the subject possesses the valid authentication data (e.g. password) of the remote interactive user already logged on to Session S;
* the subject is granted the connection right ([WINSTATION\_CONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to the specified Session S according to Session S’s “Window terminal session access” policy;
* the subject is granted the disconnection right ([WINSTATION\_DISCONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to the specified Session T according to Session T’s “Window terminal session access” policy,

then

* the Windows OS remote window terminal service requests Session T’s Windows OS window manager to disconnect Session T from the local display terminal so that no contents (i.e. window graphical interfaces) from the display areas of Session T are displayed to the local interactive user of Session T and no input from the local interactive user of Session T is accepted;
* the Windows OS remote window terminal service generates the [Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value) audit record which identifies Session T after the Session T disconnection occurs;
* the Windows OS remote window terminal service requests Session S’s Windows OS window manager to connect Session S to the local display terminal so that contents (i.e. window graphical interfaces) from the display areas of Session S are displayed to the local interactive user (who previously was seeing window graphical interfaces from Session T) and input from the local interactive user is accepted;
* the Windows OS remote window terminal service generates the [[Event ID 4778](http://support.microsoft.com/kb/947226) “A session was reconnected to a Window Station” (SE\_AUDITID\_ETW\_SESSION\_RECONNECTED\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record which identifies Session S after the Session S reconnection occurs;
* Session S becomes a local connected window terminal session;
* Session T becomes a disconnected window terminal session.

By default, an administrator possesses the disconnection right ([WINSTATION\_DISCONNECT](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to any local or remote window terminal session.

### Forced logoff of a window terminal session

If a subject requests the Windows OS remote window terminal service to send the instance of the window logon state maintaining service of a specific window terminal session a request to log off, shutdown or restart, then

* the subject must have been granted the “Permission to reset or end the network listener associated with a window terminal session, or to logoff a window terminal session” right ([WINSTATION\_RESET](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) according to the “Window terminal session access” policy for the specified window terminal session.

By default, an administrator possesses the “Permission to reset or end the network listener associated with a window terminal session, or to logoff a window terminal session” right ([WINSTATION\_RESET](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to any local or remote window terminal session.

Consequently, this Commercial Grade OS Requirement Set “3.2.2.2” requirement is met.

## Addressing 3.2.2.3 “The OS shall provide authorized administrators with the ability to specify the number of concurrent interactive sessions allowed per user”

This requirement is addressed by the Windows OS as follows.

In the justification text for addressing the Commercial Grade OS Requirement Set “3.2.1.4” requirement, we have presented a “workaround” solution for meeting the intent of the requirement.

In the “workaround” solution, we let N be the maximum number of interactive logging on sessions for a specific user account defined by an administrator. The administrator selects N Windows OS machines which are member computers of the Windows OS domain managed by the administrator.

In each of those N selected Windows OS machines, the administrator

* assigns the [SeInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx) and [SeRemoteInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx) Windows OS system access rights to the specific user account.

In each of the remaining Windows OS machines in the Windows OS domain which are not those N selected Windows OS machines, the administrator

* assigns the [SeDenyInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx) and [SeDenyRemoteInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx) Windows OS system access rights to the specific user account.

The enforcement effect of the above Windows OS system access right assignment to the specific user account clearly implies that N is the maximum number of interactive logging on sessions for the specific user account.

Finally, we note that the Windows OS system access right assignment on individual Windows OS machines is just part of the Windows OS group policy based security settings. As explained in the justification text for addressing the Commercial Grade OS Requirement Set “1.2.2.1” requirement, the Windows OS group policy based security settings for individual Windows OS machines can be managed and configured centralized by an administrator in his/her Windows OS forest/domain.

Consequently, this Commercial Grade OS Requirement Set “3.2.2.3” requirement is met.

## Addressing 3.2.2.4 “The OS shall provide authorized administrators with the ability to deny interactive session establishment based on system parameters specified in “3.2.1.6””

This requirement is addressed by the Windows OS as follows.

The possible action values which could be used to determine the action (which is the action to be taken by the instance of the window logon state maintaining service when the user’s allowable logon hours expire) due to the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy are:

* “nothing”;
* lock the display areas;
* disconnect the window terminal session where the instance of the window logon state maintaining service resides in;
* forced logoff for the already authenticated interactive user.

If a subject requests to set the action value which is used to determine the action (which is the action to be taken by the instance of the window logon state maintaining service when the user’s allowable logon hours expire) due to the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy as a per user behavior item for the user account that the subject represents, then

* the behavior item must be allowed by the administrator specified policy.

The following registry key value defining the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy action value does not allow the non privileged user to write.

* HKEY\_CURRENT\_USER\Software\Microsoft\Windows\CurrentVersion\Policies\System\LogonHoursAction.

Finally, as explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement, the following attribute is defined for a user account object.

* [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It specifies the [hours](http://technet2.microsoft.com/windowsserver/en/library/a0f007ba-b3ef-4c1f-8836-9087fe5eb08d1033.mspx?mfr=true) that the user account is allowed to be logged on in the Windows OS domain;
  + By default, a standard user does not possess the right to update this attribute value.

Consequently, this Commercial Grade OS Requirement Set “3.2.2.4” requirement is met.

## Addressing 3.2.2.5 “The OS shall provide authorized administrators with the ability to specify a time interval of session inactivity after which an (inactive) interactive session is locked”

This requirement is addressed by the Windows OS as follows.

As described in the ““Screen saver timeout” policy” section, the “screen saver timeout” policy specifies how much user input idle time must elapse before the Windows OS window manager of a specific window terminal session notifies the instance of the window logon state maintaining service of the same session about the user input inactivity timeout. The default “screen saver timeout” value is 600 seconds from the default user profile (HKEY\_USERS\.Default\Control Panel\Desktop).

By default, only an administrator is allowed to configure the “screen saver” policy.

For the “screen saver timeout” policy to be effective, it is necessary that applications are blocked to insert injected keyboard or mouse events into the input stream for being delivered to window objects so that the screen saver timeout timer is reset. Some application (such as the Windows Media Player) has its specific understandable reason to insert injected keyboard or mouse events. For example, when the Windows Media Player is playing a video movie, it does not wish its user interactive application display area to be locked automatically due to a timeout of user inactivity.

The “Block send input to reset” policy defines the value for determining whether applications are blocked to insert injected keyboard or mouse events into the input stream for being delivered to window objects.

By default, the “Block send input to reset” policy specifies “applications are not blocked to insert injected keyboard or mouse events into the input stream for being delivered to window objects”.

The following registry key value defining the “Block send input to reset” policy does not allow the non privileged user to write.

* HKEY\_LOCAL\_MACHINE\Software\Policies\Microsoft\Windows\Control Panel\Desktop\ BlockSendInputResets.

Consequently, this Commercial Grade OS Requirement Set “3.2.2.5” requirement is met.

# Meeting the “Identification and Authentication User Interface Security Audit Requirements”

In the Commercial Grade OS Requirement Set, there are 5 individual audit requirements under the heading of “Identification and Authentication User Interface Security Audit Requirements”. They are listed as “3.2.3.n”, where n = 1, 2, 3, and 4.

## Addressing 3.2.3.1 “The OS shall provide the ability to audit all user attempts to re-authenticate to the system (for the interactive session unlocking purpose)”

This requirement is addressed by the Windows OS as follows.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.3.5” requirement, the [Event ID 4624](http://support.microsoft.com/kb/947226) “An account was successfully logged on” (SE\_AUDITID\_ETW\_SUCCESSFUL\_LOGON\_value) audit record is generated when handling the user (re)authentication processing for a user account occurring in the local Windows OS machine. This audit record includes the “LogonType” element. Similarly, the [Event ID 4625](http://support.microsoft.com/kb/947226) “An account failed to log on” (SE\_AUDITID\_ETW\_LOGON\_FAILURE\_value) audit record also includes the “LogonType” element.

When the “LogonType” element indicates “Unlock (i.e. 7)” in the [Event ID 4624](http://support.microsoft.com/kb/947226) or [Event ID 4625](http://support.microsoft.com/kb/947226) audit record, it implies that the user (re)authentication processing for the user account is attempted for the purpose to unlock the display areas of the window terminal session where the user account is logged on.

## Addressing 3.2.3.2 “The OS shall provide the ability to audit all administrator attempts to terminate a user’s locked session”

This requirement is addressed by the Windows OS as follows.

As described in the ““Initiate lock on display areas” state” section and the ““Initiate lock during resume” state” section of this paper, the [Event ID 4800](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_WORKSTATION\_LOCKED) “The (workstation) display areas are locked” audit record is generated, with the following informational items when available:

* Target User Name:
* Target Domain Name:
* Target User SID:
* Target Logon ID:
* Session ID:.

As described in the following sections:

* the ““Switch user” button” section;
* the “Transitions from the “Get session confirmation from LWTS” state” section;
* the “Remote or local window terminal session disconnection” section;
* the “Remote or local window terminal session reconnection from a (distinct) source window terminal session” section,

the [Event ID 4779](http://support.microsoft.com/kb/947226) “A session was disconnected from a Window Station” (SE\_AUDITID\_ETW\_SESSION\_DISCONNECTED\_value) audit record is generated, with the following informational items when available:

* User Account Name:
* User Account Domain Name:
* Logon ID:
* Session ID:
* Client IP Address:
* Client User Name:
* Client Domain Name:.

As explained in the “Forced logoff of a window terminal session” section of this paper, the “Permission to reset or end the network listener associated with a window terminal session, or to logoff a window terminal session” right ([WINSTATION\_RESET](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) according to the “Window terminal session access” policy for the specified window terminal session is necessary to request the Windows OS remote window terminal service to send the instance of the window logon state maintaining service of a specific window terminal session a request to log off, shutdown or restart.

As explained in the following section of this paper, the “Window terminal session access” policy supports the auditing of the success event which grants the “Permission to reset or end the network listener associated with a window terminal session, or to logoff a window terminal session” right ([WINSTATION\_RESET](http://msdn.microsoft.com/en-us/library/aa383773(VS.85).aspx)) to an administrator.

* “Window terminal session specific “Window terminal session access” policy”.

The corresponding audit event record is

* [Event ID 4656](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OPEN\_HANDLE\_value) “A handle to an object was requested” for success.

Finally, in the various specific transitions to the “Welcome” state described earlier, the [Event ID 4647](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_BEGIN\_LOGOFF.id) “User initiated logoff” audit record is also generated, with the following informational items when available:

* Target User Name:
* Target Domain Name:
* Target User SID:
* Target Logon ID:.

As a result, the combined occurrences of [Event ID 4800](http://support.microsoft.com/kb/947226) and [Event ID 4779](http://support.microsoft.com/kb/947226) and the combined occurrences of [Event ID 4800](http://support.microsoft.com/kb/947226), [Event ID 4656](http://support.microsoft.com/kb/947226), and [Event ID 4647](http://support.microsoft.com/kb/947226) meet the intents of this Commercial Grade OS Requirement Set “3.2.3.2” requirement.

## Addressing 3.2.3.3 “The OS shall provide the ability to audit any attempt to exceed the maximum number of concurrent interactive sessions by a user”

This requirement is addressed by the Windows OS as follows.

### Logging off audit event record generation

As explained earlier in the “User authentication” section, an access token is basically the “security context” associated with a process or a thread on a local Windows OS machine. While the Windows OS security reference monitor (SRM) in the kernel mode has the responsibility to manage and maintain access tokens, it is the Windows OS Authentication Service, which is responsible to request the Windows OS SRM to create an access token after it successfully authenticates a (local or remote) user for a specific user account. The created access token then represents all the activities of the logged on user account on the local Windows OS machine.

When the Windows OS security reference monitor (SRM) detects that there is no more reference to a specific access token, it informs the Windows OS Authentication Service using the “Lsap Logon Session Deleted” LPC-based internal command about the access token, before releasing the SRM resources associated with the access token. The Windows OS Authentication Service cleans up the resources that it has been maintaining for the specific logged on user account, represented by the access token. After the cleaning up, the Windows OS Authentication Service generates the [[Event ID 4634](http://support.microsoft.com/kb/947226) “An account was logged off”](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) ([SE\_AUDITID\_ETW\_LOGOFF\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record which identifies the logged on user account which is currently logging off, due to the zero count of the references to the access token representing the user account.

### Summary of the Windows OS justification for meeting “3.2.3.3”

We recall from the justification text for addressing the Commercial Grade OS Requirement Set “3.2.1.4” requirement that N is the maximum number of interactive logging on sessions for a specific user account defined by an administrator. In each of the N Windows OS machines selected by the administrator, the administrator

* assigns the [SeInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx) and [SeRemoteInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx) Windows OS system access rights to the specific user account.

In each of the remaining Windows OS machines in the Windows OS domain which are not those N selected Windows OS machines, the administrator

* assigns the [SeDenyInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx) and [SeDenyRemoteInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx) Windows OS system access rights to the specific user account.

Therefore, when the following event pattern is detected in the combined audit record histories collected from all the Windows OS machines belonging to the Windows OS forest/domain, it implies that an attempt to exceed the maximum number of concurrent interactive sessions for a specific user account has occurred.

* The occurrence of an [Event ID 4624](http://support.microsoft.com/kb/947226) “An account was successfully logged on” (SE\_AUDITID\_ETW\_SUCCESSFUL\_LOGON\_value) audit record identifying the specific user account in each of the N Windows OS machines selected by the administrator.
* The occurrence of an [Event ID 4625](http://support.microsoft.com/kb/947226) “An account failed to log on” (SE\_AUDITID\_ETW\_LOGON\_FAILURE\_value) audit record identifying the specific user account with the following “Failure Reason”:
  + “The user has not been granted the requested logon type at this machine” due to [STATUS\_LOGON\_TYPE\_NOT\_GRANTED](http://msdn.microsoft.com/en-us/library/cc704588.aspx).

in any other Windows OS machine in the Windows OS forest/domain but not the N Windows OS machines selected by the administrator.

* There is no occurrence of an [[Event ID 4634](http://support.microsoft.com/kb/947226) “An account was logged off”](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) ([SE\_AUDITID\_ETW\_LOGOFF\_value)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) audit record identifying the specific user account in any of the N Windows OS machines selected by the administrator
  + after one of the above occurrence of the corresponding [Event ID 4624](http://support.microsoft.com/kb/947226) “successful logon” audit record; and
  + before the above occurrence of the [Event ID 4625](http://support.microsoft.com/kb/947226) “logon failure” audit record.

Consequently, this Commercial Grade OS Requirement Set “3.2.3.3” requirement is addressed.

## Addressing 3.2.3.4 “The OS shall provide the ability to audit all user attempts that violate the restrictions specified in “3.2.1.6””

This requirement is addressed by the Windows OS as follows.

As described in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.3.5” requirement, the [Event ID 4625](http://support.microsoft.com/kb/947226) “An account failed to log on” (SE\_AUDITID\_ETW\_LOGON\_FAILURE\_value) audit record is generated when handling the user authentication processing for a user account. This audit record identifies a specific user account with the following “Failure Reason”:

* “Account logon time restriction violation” due to [STATUS\_INVALID\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/cc704588.aspx) and [STATUS\_ACCOUNT\_RESTRICTION](http://msdn.microsoft.com/en-us/library/cc704588.aspx).

Additionally, as described in the ““Notify logon subscribers” state” section, while still remaining in the “Notify logon subscribers” state, if the action for the already authenticated interactive user who is not an administrator is due to a non-“nothing” action of the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy, then

* the instance of the window logon state maintaining service generates the [[Event ID 4002](http://technet.microsoft.com/en-us/library/cc733989.aspx) “The logon hours restriction policy is applied to the logged on user. The user's session has been locked, disconnected or logged off depending on the policy setting” (EVENT\_LOGON\_HOURS\_POLICY\_APPLIED)](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=644&EvtSrc=Security&LCID=1033) soft audit record which identifies the already authenticated interactive user for the soft audit storage.

The soft audit storage is the audit storage that does not cause the local machine to prevent subsequent auditable events, except those taken by an administrator, to occur when the audit storage is full (i.e. reaching its maximum size).

Consequently, this Commercial Grade OS Requirement Set “3.2.3.4” requirement is met.

# Meeting the “Cryptographic Support Operations Functional Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual functional requirement under the heading of “Cryptographic Support Operations Functional Requirements”. It is listed as “4.1.1.1”.

## Addressing 4.1.1.1 “The OS shall provide a specific list of cryptographic services to applications”

The Commercial Grade OS Requirement Set requires the following list of cryptographic services to be provided to applications:

1. Encryption/Decryption;
2. Digital Signature;
3. Hashing;
4. Cryptographic Key Agreement;
5. Randomization;
6. Key Management.

This requirement is addressed by the Windows OS as follows.

The Windows OS provides the FIPS 140-2 validated [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) for applications to use their cryptographic services in the user mode and the kernel mode respectively. The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received FIPS-140-2 (Cert # 892 and Cert # 1008) and the Windows OS kernel security device driver (ksecdd.sys) has received FIPS-140-2 (Cert # 891 and Cert # 1007).

The details of the above required cryptographic services are discussed in the justification text for addressing the Commercial Grade OS Requirement Set “4.2.1.2”, “4.3.1.1”, and “3.4.4.1.n” requirements, where n = 1, 2, 3, 4, 5, 6, 7, and 8, below in this paper.

# Meeting the “Cryptographic Support Operations Management Requirements”

In the Commercial Grade OS Requirement Set, there is no individual management requirement under the heading of “Cryptographic Support Operations Management Requirements”.

# Meeting the “Cryptographic Support Operations Audit Requirements”

In the Commercial Grade OS Requirement Set, there is no individual audit requirement under the heading of “Cryptographic Support Operations Audit Requirements”.

# Meeting the “Cryptographic Support Capabilities Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 2 individual functional requirements under the heading of “Cryptographic Support Capabilities Security Functional Requirements”. They are listed as “4.2.1.n”, where n = 1 and 2.

## Addressing 4.2.1.1 “The OS shall provide FIPS-approved cryptographic algorithms and modes of operations, implemented in a crypto module that is FIPS 140-2 Security Level 2 validated”

The Commercial Grade OS Requirement Set requires the evidence of the conformance with the following list of FIPS 140-2 Security Level 2 validation areas in the FIPS 140-2 validated crypto module:

* Cryptographic Module Specification;
* Cryptographic Module Ports and Interfaces;
* Roles, Services and Authentication;
* Finite State Model;
* Cryptographic Key Management;
* Self-Tests;
* Design Assurance.

This requirement is addressed by the Windows OS as follows.

The Windows OS provides the FIPS 140-2 (overall Security Level 1) validated [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) for applications to use their cryptographic services in the user mode and the kernel mode respectively. The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received FIPS-140-2 (Cert # 892 and Cert # 1008) and the Windows OS kernel security device driver (ksecdd.sys) has received FIPS-140-2 (Cert # 891 and Cert # 1007).

### FIPS 140-2 “Cryptographic Module Specification” validation area

In the case of the “Cryptographic Module Specification” validation area, its assertions as specified in the [[Derived Test Requirements for FIPS 140-2, Security Requirements for Cryptographic Modules](http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/fips1402DTR.pdf)] for Security Level 1 and for Security Level 2 are identical. Therefore, the FIPS 140-2 overall Security Level 1 statuses for the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) suffice for the “Cryptographic Module Specification” validation area with respect to Security Level 2 also.

### FIPS 140-2 “Cryptographic Module Ports and Interfaces” validation area

In the case of the “Cryptographic Module Ports and Interfaces” validation area, its assertions as specified in the [[Derived Test Requirements for FIPS 140-2, Security Requirements for Cryptographic Modules](http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/fips1402DTR.pdf)] for Security Level 1 and for Security Level 2 are identical. Therefore, the FIPS 140-2 overall Security Level 1 statuses for the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) suffice for the “Cryptographic Module Ports and Interfaces” validation area with respect to Security Level 2 also.

### FIPS 140-2 “Roles, Services and Authentication” validation area

In the case of the “Roles, Services and Authentication” validation area, its assertions as specified in [[Derived Test Requirements for FIPS 140-2, Security Requirements for Cryptographic Modules](http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/fips1402DTR.pdf)] for Security Level 1 and for Security Level 2 are identical, except the following FIPS 140-2 Security Level 2 assertions.

* AS03.16: (Levels 2, 3, and 4)
  + Depending on the security level, the cryptographic module shall perform at least one of the following mechanisms to control access to the module: role-based authentication or identity-based authentication.
* AS03.17: (Level 2)
  + If role-based authentication mechanisms are supported by the cryptographic module, the module shall require that one or more roles either be implicitly or explicitly selected by the operator and shall authenticate the assumption of the selected role (or set of roles).
* AS03.18: (Level 2)
  + If the cryptographic module permits an operator to change roles, then the module shall authenticate the assumption of any role that was not previously authenticated.
* AS03.22: (Levels 2, 3, and 4)
  + Authentication data within the cryptographic module shall be protected against unauthorized disclosure, modification, and substitution.
* AS03.24: (Levels 2, 3, and 4)
  + The strength of the authentication mechanism shall conform to the specifications of AS03.25 and AS03.26.
* AS03.25: (Levels 2, 3, and 4)
  + For each attempt to use the authentication mechanism, the probability shall be less than one in 1,000,000 that a random attempt will succeed or a false acceptance will occur (e.g., guessing a password or PIN, false acceptance error rate of a biometric device, or some combination of authentication methods).
* AS03.26: (Levels 2, 3, and 4)
  + For multiple attempts to use the authentication mechanism during a one-minute period, the probability shall be less than one in 100,000 that a random attempt will succeed or a false acceptance will occur.
* AS03.27: (Levels 2, 3, and 4)
  + Feedback of authentication data to an operator shall be obscured during authentication (e.g., no visible display of characters when entering a password).
* AS03.28: (Levels 2, 3, and 4)
  + Feedback provided to an operator during an attempted authentication shall not weaken the strength of the authentication mechanism.
* AS03.31: (Level 2)
  + A cryptographic module shall employ role-based authentication to control access to the module.

As a crypto module, neither the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) nor the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) performs authentication of users. As explained in the “

Meeting the “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Functional Requirements”” section of this paper, it is the Windows OS authentication service that actually performs an identity-based authentication mechanism in the critical path for controlling access to the Windows OS, and hence to the Windows OS crypto modules. As a result, the AS03.16 assertion is satisfied implicitly.

The AS03.17 assertion is moot as the Windows OS authentication service performs an identity-based authentication mechanism and not a role-based authentication mechanism for controlling access to the Windows OS.

The AS03.18 assertion is moot as the Windows OS does not support a logged on user to change roles based on the user security attribute association, as explained in the following section of this paper.

* “Addressing 3.1.1.8 “The OS shall associate a specific list of user security attributes with subjects acting on behalf of that user””.

The AS03.22 assertion is satisfied due to the Windows OS capabilities as explained in the “Password lifecycle” section of this paper.

The AS03.24, AS03.25 and AS03.28 assertions are satisfied due to the Windows OS capabilities as explained in the following section of this paper.

* “Addressing 3.1.1.2 “The OS shall be able to support passwords up to 32 characters in length, consisting of any combination of upper and lower case letters, numbers, and punctuations””.

The AS03.24, AS03.26 and AS03.28 assertions are satisfied due to the Windows OS capabilities as explained in the following section of this paper.

* “Addressing 3.1.1.3 “The OS authentication mechanism shall provide a specific set of capabilities””.

The AS03.27 assertion is satisfied due to the Windows OS capabilities as explained in the following section of this paper.

* “Addressing 3.1.1.4 “The OS shall provide only obscured feedback to the user while the authentication is in progress””.

The AS03.31 assertion is irrelevant because

* the Windows OS authentication service performs an identity-based authentication mechanism and not a role-based authentication mechanism for controlling access to the Windows OS;
* the “FIPS 140-2, Security Requirements for Cryptographic Modules” states that “Security Level 3 requires identity-based authentication mechanisms, enhancing the security provided by the role-based authentication mechanisms specified for Security Level 2”, which implies that the identity-based authentication mechanism of the Windows OS authentication service suffices for Security Level 2.

Therefore, the FIPS 140-2 overall Security Level 1 statuses for the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) suffice for the “Roles, Services and Authentication” validation area with respect to Security Level 2 also, because of the above justifications for the excepted assertions.

### FIPS 140-2 “Finite State Model” validation area

In the case of the “Finite State Model” validation area, its assertions as specified in the [[Derived Test Requirements for FIPS 140-2, Security Requirements for Cryptographic Modules](http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/fips1402DTR.pdf)] for Security Level 1 and for Security Level 2 are identical. Therefore, the FIPS 140-2 overall Security Level 1 statuses for the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) suffice for the “Finite State Model” validation area with respect to Security Level 2 also.

### FIPS 140-2 “Cryptographic Key Management” validation area

In the case of the “Cryptographic Key Management” validation area, its assertions as specified in the [[Derived Test Requirements for FIPS 140-2, Security Requirements for Cryptographic Modules](http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/fips1402DTR.pdf)] for Security Level 1 and for Security Level 2 are identical. Therefore, the FIPS 140-2 overall Security Level 1 statuses for the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) suffice for the “Cryptographic Key Management” validation area with respect to Security Level 2 also.

### FIPS 140-2 “Self-Tests” validation area

In the case of the “Self-Tests” validation area, its assertions as specified in the [[Derived Test Requirements for FIPS 140-2, Security Requirements for Cryptographic Modules](http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/fips1402DTR.pdf)] for Security Level 1 and for Security Level 2 are identical. Therefore, the FIPS 140-2 overall Security Level 1 statuses for the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) suffice for the “Self-Tests” validation area with respect to Security Level 2 also.

### FIPS 140-2 “Design Assurance” validation area

In the case of the “Design Assurance” validation area, its assertions as specified in the [[Derived Test Requirements for FIPS 140-2, Security Requirements for Cryptographic Modules](http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/fips1402DTR.pdf)] for Security Level 1 and for Security Level 2 are identical, except the following FIPS 140-2 Security Level 2 assertions.

* AS10.04: (Levels 2, 3, and 4)
  + In addition to the requirements of Security Level 1, documentation shall specify the procedures required for maintaining security while distributing and delivering versions of the cryptographic module to authorized operators.
* AS10.09: (Levels 2, 3, and 4)
  + In addition to the requirements for Security Level 1, the following requirement of AS10.10 shall apply to cryptographic modules for Security Level 2.
* AS10.10: (Levels 2, 3, and 4)
  + Documentation shall specify a functional specification that informally describes the cryptographic module, the external ports and interfaces of the module, and the purpose of the interfaces.

The AS10.04 assertion is satisfied because the integrity of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) is ensured as explained in the following section of this paper.

* “Addressing 5.3.1.3 “The OS shall verify during initial startup the integrity of executable code that implements access control and cryptographic functionality through the use of the OS system provided cryptographic services””.

The authorized administrators only need to manually inspect that the [IMAGE\_DLLCHARACTERISTICS\_FORCE\_INTEGRITY](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) flag is set in the [DllCharacteristics](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) field of the [IMAGE\_OPTIONAL\_HEADER](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) of bcrypt.dll as a Windows OS image file.

The AS10.10 assertion is satisfied because of the publication of the documentation about the interfaces and external ports of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) in the following links:

* <http://msdn.microsoft.com/en-us/library/aa376210(VS.85).aspx>;
* <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1008.pdf>;
* <http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf>.

Therefore, the FIPS 140-2 overall Security Level 1 statuses for the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) suffice for the “Design Assurance” validation area with respect to Security Level 2 also, because of the above justifications for the excepted assertions.

Consequently, this Commercial Grade OS Requirement Set “4.2.1.1” requirement is addressed.

## Addressing 4.2.1.2 “A specific list of algorithms shall be provided for the corresponding cryptographic services”

The Commercial Grade OS Requirement Set requires the following list of algorithms.

1. For encryption/decryption, either
   1. Triple Data Encryption Algorithm (TDEA) used in FIPS-approved modes of operation and cryptographic key size of at least 168 bits (three independent keys);
   2. Advanced Encryption Standard (AES) used in FIPS-approved modes of operation and cryptographic key size of at least 128 bits;
2. For digital signature, either
   1. Digital Signature Algorithm (DSA) with a key size (modulus) of at least 1028 bits;
   2. RSA Digital Signature Algorithm (rDSA) with a key size of at least 2048 bits;
   3. Elliptic Curve Digital Signature Algorithm (ECDSA) using only the NIST-curves with a key size of at least 256 bits, using NIST P curves;
3. For hashing, either
   1. SHA-1;
   2. SHA-256;
   3. SHA-512;
4. For key agreement, either
   1. Finite Field-based key agreement algorithm and cryptographic key sizes (modulus) of at least 2048 bits as specified in [FIPS 140-2 Annex D];
   2. Elliptic Curve-based key agreement algorithm and cryptographic key size of at least 256 bits as specified in [FIPS 140-2 Annex D].

This requirement is addressed by the Windows OS as follows.

### Encryption/Decryption

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following TDEA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 656 for TECB (e/d; KO 1,2), TCBC (e/d; KO 1,2), TCFB8 (e/d; KO 1,2).

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has received the following TDEA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 656 for TECB (e/d; KO 1,2), TCBC (e/d; KO 1,2), TCFB8 (e/d; KO 1,2).

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following AES algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 739 for ECB (e/d; 128,192,256), CBC (e/d; 128,192,256), CFB8 (e/d; 128,192,256);
* Cert # 757 for CCM (KS: 128 , 192 , 256);
* Cert # 756 for CCM (KS: 128 , 192 , 256).

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has received the following AES algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 739 for ECB (e/d; 128,192,256), CBC (e/d; 128,192,256), CFB8 (e/d; 128,192,256);
* Cert # 757 for CCM (KS: 128 , 192 , 256);
* Cert # 756 for CCM (KS: 128 , 192 , 256).

### Digital signature

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following DSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 283 for KEYGEN(Y) MOD(1024), SIG(gen) MOD(1024), SIG(ver) MOD(1024);
* Cert # 284 for KEYGEN(Y) MOD(1024), SIG(gen) MOD(1024), SIG(ver) MOD(1024).

Note that [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct DSA algorithm validation for key size (modulus) greater than 1024 bits.

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following rDSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 357 for
  + ALG[RSASSA-PKCS1\_V1\_5] SIG(gen), SIG(ver), 1024 , 1536, 2048, 3072, 4096;
  + ALG[RSASSA-PSS] SIG(gen), SIG(ver), 1024 , 1536, 2048, 3072, 4096;
* Cert # 358 for
  + ALG[RSASSA-PKCS1\_V1\_5] SIG(gen), SIG(ver), 1024 , 1536, 2048, 3072, 4096;
  + ALG[RSASSA-PSS] SIG(gen), SIG(ver), 1024 , 1536, 2048, 3072, 4096;
* Cert # 353 for
  + ALG[ANSIX9.31] Key(gen) (MOD: 1024, 1536, 2048, 3072, 4096; PubKey Values: 65537).

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has received the following rDSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 357 for
  + ALG[RSASSA-PKCS1\_V1\_5] SIG(gen), SIG(ver), 1024 , 1536, 2048, 3072, 4096;
  + ALG[RSASSA-PSS] SIG(gen), SIG(ver), 1024 , 1536, 2048, 3072, 4096;
* Cert # 358 for
  + ALG[RSASSA-PKCS1\_V1\_5] SIG(gen), SIG(ver), 1024 , 1536, 2048, 3072, 4096;
  + ALG[RSASSA-PSS] SIG(gen), SIG(ver), 1024 , 1536, 2048, 3072, 4096;
* Cert # 353 for
  + ALG[ANSIX9.31] Key(gen)(MOD: 1024, 1536, 2048, 3072, 4096; PubKey Values: 65537).

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following ECDSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 83 for
  + PKG: CURVES(P-256, P-384, P-512);
  + SIG(gen): CURVES(P-256, P-384, P-512);
  + SIG(ver): CURVES(P-256, P-384, P-512);
* Cert # 82 for
  + PKG: CURVES(P-256, P-384, P-512);
  + SIG(gen): CURVES(P-256, P-384, P-512);
  + SIG(ver): CURVES(P-256, P-384, P-512).

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has received the following ECDSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 83 for
  + PKG: CURVES(P-256, P-384, P-512);
  + SIG(gen): CURVES(P-256, P-384, P-512);
  + SIG(ver): CURVES(P-256, P-384, P-512);
* Cert # 82 for
  + PKG: CURVES(P-256, P-384, P-512);
  + SIG(gen): CURVES(P-256, P-384, P-512);
  + SIG(ver): CURVES(P-256, P-384, P-512).

### Hashing

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following SHA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 753 for
  + SHA-1 (BYTE-only);
  + SHA-256 (BYTE-only);
  + SHA-384 (BYTE-only);
  + SHA-512 (BYTE-only).

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has received the following SHA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 753 for
  + SHA-1 (BYTE-only);
  + SHA-256 (BYTE-only);
  + SHA-384 (BYTE-only);
  + SHA-512 (BYTE-only).

### Key agreement

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) supports the key pair generation of the Diffie-Hellman Finite Field-based key agreement algorithm for key agreement with the key size greater than or equal to 384 bits, but less than or equal to 4096 bits through its [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) interface. The use of this key agreement algorithm is exercised in the [BCryptSecretAgreement()](http://msdn.microsoft.com/en-us/library/aa375496(VS.85).aspx) and the [BCryptDeriveKey()](http://msdn.microsoft.com/en-us/library/aa375393(VS.85).aspx) interfaces.

Note that, currently, [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct validation tests for implementations of the Diffie-Hellman Finite Field-based key agreement algorithm for key agreement.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) also supports the key pair generation of the Diffie-Hellman Finite Field-based key agreement algorithm for key agreement with the key size greater than or equal to 384 bits, but less than or equal to 4096 bits through its [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) interface. The use of this key agreement algorithm is exercised in the [BCryptSecretAgreement()](http://msdn.microsoft.com/en-us/library/aa375496(VS.85).aspx) and the [BCryptDeriveKey()](http://msdn.microsoft.com/en-us/library/aa375393(VS.85).aspx) interfaces.

Note that, currently, [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct validation tests for implementations of the Diffie-Hellman Finite Field-based key agreement algorithm for key agreement.

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) supports the key pair generation of the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm for key agreement with the following NIST P curves and corresponding key sizes:

* P-256 and 256 bits key size;
* P-384 and 384 bits key size;
* P-512 and 512 bits key size.

through its [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) interface. The use of this key agreement algorithm is exercised in the [BCryptSecretAgreement()](http://msdn.microsoft.com/en-us/library/aa375496(VS.85).aspx) and the [BCryptDeriveKey()](http://msdn.microsoft.com/en-us/library/aa375393(VS.85).aspx) interfaces.

Note that, currently, [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct validation tests for implementations of the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm for key agreement.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) also supports the key pair generation of the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm for key agreement with the following NIST P curves and corresponding key sizes:

* P-256 and 256 bits key size;
* P-384 and 384 bits key size;
* P-512 and 512 bits key size.

through its [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) interface. The use of this key agreement algorithm is exercised in the [BCryptSecretAgreement()](http://msdn.microsoft.com/en-us/library/aa375496(VS.85).aspx) and the [BCryptDeriveKey()](http://msdn.microsoft.com/en-us/library/aa375393(VS.85).aspx) interfaces.

Note that, currently, [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct validation tests for implementations of the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm for key agreement.

Consequently, this Commercial Grade OS Requirement Set “4.2.1.2” requirement is met.

# Meeting the “Cryptographic Support Capabilities Management Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual Management requirement under the heading of “Cryptographic Support Capabilities Management Requirements”. It is listed as “4.2.2.1”.

## Addressing 4.2.2.1 “The OS shall provide the authorized administrators the ability to select (when more than one algorithm is available) the appropriate algorithm for encryption/decryption, digital signature, hashing, and key agreement”

This requirement is addressed by the Windows OS as follows.

The cryptographic services provided by the Windows OS crypto modules, namely the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys), are meant to be used by multiple applications running on the Windows OS. Due to the support of their unique interoperability requirements, these applications are unlikely to share the same algorithm requirements. As a result, all the cryptographic services implemented in the Windows OS crypto modules are available to applications. The Windows OS does not provide a mechanism for an authorized administrator to limit the selections of available cryptographic services implemented in the Windows OS crypto modules.

In fact, the Windows OS provides every subject, and not just the authorized administrators, the ability to select the appropriate algorithm for encryption/decryption, digital signature, hashing, and key agreement in their applications.

Consequently, this Commercial Grade OS Requirement Set “4.2.2.1” requirement is addressed.

# Meeting the “Cryptographic Support Capabilities Audit Requirements”

In the Commercial Grade OS Requirement Set, there is no individual audit requirement under the heading of “Cryptographic Support Capabilities Audit Requirements”.

# Meeting the “Cryptographic Support Randomization Functional Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual functional requirement under the heading of “Cryptographic Support Randomization Functional Requirements”. It is listed as “4.3.1.1”.

## Addressing 4.3.1.1 “The OS crypto module shall provide Random Number Generation (RNG) services in accordance with a FIPS-Approved RNG listed in [FIPS 140-2 Annex C] composed of a specific method”

The Commercial Grade OS Requirement Set requires the selection of one of the following RNG method that complies with the tests specified in [NIST SP 800-90].

1. At least one independent hardware-generated input (noise source) combined with a FIPS-approved cryptographic hashing function;
2. At least one independent software-generated input (noise source) combined with a FIPS-approved cryptographic hashing function;
3. A combination of at least one independent hardware-generated input (noise source) combined with a mixing function and at least one independent software-generated input (noise source) combined with a FIPS-approved cryptographic hashing function.

This requirement is addressed by the Windows OS as follows.

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) provides the Random Number Generation (RNG) services in its [BCryptGenRandom()](http://msdn.microsoft.com/en-us/library/aa375458(VS.85).aspx) interface. The caller subject may select one of the following three algorithms, all of which compose of one independent software-generated input (noise source) combined with a FIPS-approved cryptographic hashing function (i.e. the RNG method 2) above):

* [BCRYPT\_RNG\_FIPS186\_DSA\_ALGORITHM (FIPS186DSARNG)](http://msdn.microsoft.com/en-us/library/aa375534.aspx)
  + This is the random-number generator algorithm from FIPS 186-2 DSA (Digital Signature Algorithm);
  + The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following corresponding RNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for this algorithm:
    - Cert # 435 for
      * FIPS 186-2 [(x-Change Notice),(SHA-1)];
      * FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)];
* [BCRYPT\_RNG\_DUAL\_EC\_ALGORITHM (DUALECRNG)](http://msdn.microsoft.com/en-us/library/aa375534.aspx)
  + This is the NIST SP 800-90 dual elliptic curve deterministic random-number generator (EC\_DRGB) algorithm.
  + The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for this EC\_DRGB algorithm.
* [BCRYPT\_RNG\_ALGORITHM (RNG)](http://msdn.microsoft.com/en-us/library/aa375534.aspx)
  + This is the NIST SP 800-90 AES counter mode deterministic random-number generator (AES CTR\_DRBG) algorithm, available on the Windows OS, starting with Windows Vista SP1 and Windows Server 2008.
  + The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for this AES CTR\_DRBG algorithm.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) provides the Random Number Generation (RNG) services in its [BCryptGenRandom()](http://msdn.microsoft.com/en-us/library/aa375458(VS.85).aspx) interface. The caller subject may select one of the following three algorithms, all of which compose of one independent software-generated input (noise source) combined with a FIPS-approved cryptographic hashing function:

* [BCRYPT\_RNG\_FIPS186\_DSA\_ALGORITHM (FIPS186DSARNG)](http://msdn.microsoft.com/en-us/library/aa375534.aspx)
  + This is the random-number generator algorithm from FIPS 186-2 DSA (Digital Signature Algorithm);
  + The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has received the following corresponding RNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for this algorithm:
    - Cert # 435 for
      * FIPS 186-2 [(x-Change Notice),(SHA-1)];
      * FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)];
* [BCRYPT\_RNG\_DUAL\_EC\_ALGORITHM (DUALECRNG)](http://msdn.microsoft.com/en-us/library/aa375534.aspx)
  + This is the NIST SP 800-90 dual elliptic curve deterministic random-number generator (EC\_DRGB) algorithm.
  + The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for this EC\_DRGB algorithm.
* [BCRYPT\_RNG\_ALGORITHM (RNG)](http://msdn.microsoft.com/en-us/library/aa375534.aspx)
  + This is the NIST SP 800-90 AES counter mode deterministic random-number generator (AES CTR\_DRBG) algorithm.
  + The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for this AES CTR\_DRBG algorithm.

Except the fact that neither the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) nor the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has yet received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for their implementations of the NIST SP 800-90 EC\_DRGB and AES CTR\_DRBG algorithms, this Commercial Grade OS Requirement Set “4.3.1.1” requirement is otherwise met.

# Meeting the “Cryptographic Support Randomization Management Requirements”

In the Commercial Grade OS Requirement Set, there is no individual management requirement under the heading of “Cryptographic Support Randomization Management Requirements”.

# Meeting the “Cryptographic Support Randomization Audit Requirements”

In the Commercial Grade OS Requirement Set, there is no individual audit requirement under the heading of “Cryptographic Support Randomization Audit Requirements”.

# Meeting the “Cryptographic Support Cryptographic Key Management Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 8 individual functional requirements under the heading of “Cryptographic Support Cryptographic Key Management Security Functional Requirements”. They are listed as “4.4.1.n”, where n = 1, 2, 3, 4, 6, 7, and 8.

## Addressing 4.4.1.1 “The OS crypto module shall generate symmetric cryptographic keys using a random number generator as specified in “4.3.1.1””

This requirement is addressed by the Windows OS as follows.

In the programming model for the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys), the [BCryptGenerateSymmetricKey()](http://msdn.microsoft.com/en-us/library/aa375453(VS.85).aspx) interface is used to input a secret to create a key object for use with a symmetrical key encryption algorithm.

As long as the [BCryptGenRandom()](http://msdn.microsoft.com/en-us/library/aa375458(VS.85).aspx) interface, mentioned in the justification text for addressing the Commercial Grade OS Requirement Set “4.3.1.1” requirement, is used to generate a random number as the secret for [BCryptGenerateSymmetricKey()](http://msdn.microsoft.com/en-us/library/aa375453(VS.85).aspx) interface, the desired effects, required in this Commercial Grade OS Requirement Set “4.4.1.1” requirement, are achieved.

## Addressing 4.4.1.2 “The OS crypto module shall generate asymmetric cryptographic keys using a domain parameter generator and a random number generator as specified in “4.3.1.1””

This requirement is addressed by the Windows OS as follows.

### DSA algorithm

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following DSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 283 for KEYGEN(Y) MOD(1024);
* Cert # 284 for KEYGEN(Y) MOD(1024).

[NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) accepts Microsoft’s vendor-affirmed prerequisite NIST SP 800-90 RNG implementation that underlies the DSA asymmetric cryptographic key generation as the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation.

### rDSA algorithm

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following rDSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 353 for ALG[ANSIX9.31] Key(gen) (MOD: 1024, 1536, 2048, 3072, 4096; PubKey Values: 65537).

[NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) accepts Microsoft’s vendor-affirmed prerequisite NIST SP 800-90 RNG implementation that underlies the rDSA asymmetric cryptographic key generation as the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has received the following rDSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 353 for ALG[ANSIX9.31] Key(gen) (MOD: 1024, 1536, 2048, 3072, 4096; PubKey Values: 65537).

[NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) accepts Microsoft’s vendor-affirmed prerequisite NIST SP 800-90 RNG implementation that underlies the rDSA asymmetric cryptographic key generation as the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation.

### ECDSA algorithm

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following ECDSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 83 for
  + PKG: CURVES(P-256, P-384, P-512);
* Cert # 82 for
  + PKG: CURVES(P-256, P-384, P-512).

[NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) accepts Microsoft’s vendor-affirmed prerequisite NIST SP 800-90 RNG implementation that underlies the ECDSA asymmetric cryptographic key generation as the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has received the following ECDSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 83 for
  + PKG: CURVES(P-256, P-384, P-512);
* Cert # 82 for
  + PKG: CURVES(P-256, P-384, P-512).

[NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) accepts Microsoft’s vendor-affirmed prerequisite NIST SP 800-90 RNG implementation that underlies the ECDSA asymmetric cryptographic key generation as the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation.

### Diffie-Hellman Finite Field-based key agreement algorithm

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) supports the key pair generation of the Diffie-Hellman Finite Field-based key agreement algorithm for key agreement with the key size greater than or equal to 384 bits, but less than or equal to 4096 bits through its [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) and [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) interfaces.

Note that, currently, [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct validation tests for implementations of the Diffie-Hellman Finite Field-based key agreement algorithm for key agreement.

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation that underlies the Diffie-Hellman Finite Field-based key agreement algorithm asymmetric cryptographic key generation.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) supports the key pair generation of the Diffie-Hellman Finite Field-based key agreement algorithm for key agreement with the key size greater than or equal to 384 bits, but less than or equal to 4096 bits through its [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) and [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) interfaces.

Note that, currently, [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct validation tests for implementations of the Diffie-Hellman Finite Field-based key agreement algorithm for key agreement.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation that underlies the Diffie-Hellman Finite Field-based key agreement algorithm asymmetric cryptographic key generation.

### EC Diffie-Hellman Elliptic Curve-based key agreement algorithm

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) supports the key pair generation of the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm for key agreement with the following NIST P curves and corresponding key sizes:

* P-256 and 256 bits key size;
* P-384 and 384 bits key size;
* P-512 and 512 bits key size.

through its [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) and [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) interfaces.

Note that, currently, [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct validation tests for implementations of the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm for key agreement.

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation that underlies the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm asymmetric cryptographic key generation.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) supports the key pair generation of the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm for key agreement with the following NIST P curves and corresponding key sizes:

* P-256 and 256 bits key size;
* P-384 and 384 bits key size;
* P-512 and 512 bits key size.

through its [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) and [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) interfaces.

Note that, currently, [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct validation tests for implementations of the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm for key agreement.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation that underlies the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm asymmetric cryptographic key generation.

Except the fact that neither the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) nor the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has yet received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for their implementations of the NIST SP800-90 AES CTR\_DRBG algorithm that underlies the following asymmetric key algorithms, this Commercial Grade OS Requirement Set “4.4.1.2” requirement is otherwise met.

* The DSA algorithm;
* The rDSA algorithm;
* The ECDSA algorithm;
* The Diffie-Hellman Finite Field-based key agreement algorithm;
* The EC Diffie-Hellman Elliptic Curve-based key agreement algorithm.

## Addressing 4.4.1.3 “The OS crypto module shall generate cryptographic keys such that generated key strength shall be equivalent to, or greater than, a symmetric key strength of 128 bits using conservative estimates as specified in [NIST SP 800-57 Section 5.6.1]”

This requirement is addressed by the Windows OS as follows.

Table 2 in NIST SP 800-57 Section 5.6.1 “Comparable Algorithm Strengths” specifies the following.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Bits of security** | **Symmetric key algorithms** | **FFC (e.g. DSA, DH)** | **IFC (e.g. RSA)** | **ECC (e.g. DCDSA)** |
| 128 | AES-128 | L = 3072  N = 256 | k = 3072 | f = 256-383 |

### AES

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following AES algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 739 for ECB (e/d; 128,192,256), CBC (e/d; 128,192,256), CFB8 (e/d; 128,192,256);
* Cert # 757 for CCM (KS: 128 , 192 , 256);
* Cert # 756 for CCM (KS: 128 , 192 , 256).

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has received the following AES algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 739 for ECB (e/d; 128,192,256), CBC (e/d; 128,192,256), CFB8 (e/d; 128,192,256);
* Cert # 757 for CCM (KS: 128 , 192 , 256);
* Cert # 756 for CCM (KS: 128 , 192 , 256).

### Diffie-Hellman Finite Field-based key agreement algorithm

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) supports the key pair generation of the Diffie-Hellman Finite Field-based key agreement algorithm for key agreement with the key size greater than or equal to 384 bits, but less than or equal to 4096 bits through its [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) and [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) interfaces.

Note that, currently, [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct validation tests for implementations of the Diffie-Hellman Finite Field-based key agreement algorithm for key agreement.

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation that underlies the Diffie-Hellman Finite Field-based key agreement algorithm asymmetric cryptographic key generation.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) supports the key pair generation of the Diffie-Hellman Finite Field-based key agreement algorithm for key agreement with the key size greater than or equal to 384 bits, but less than or equal to 4096 bits through its [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) and [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) interfaces.

Note that, currently, [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct validation tests for implementations of the Diffie-Hellman Finite Field-based key agreement algorithm for key agreement.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation that underlies the Diffie-Hellman Finite Field-based key agreement algorithm asymmetric cryptographic key generation.

### rDSA algorithm

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following rDSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 353 for ALG[ANSIX9.31] Key(gen) (MOD: 1024, 1536, 2048, 3072, 4096; PubKey Values: 65537).

[NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) accepts Microsoft’s vendor-affirmed prerequisite NIST SP 800-90 RNG implementation that underlies the rDSA asymmetric cryptographic key generation as the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has received the following rDSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 353 for ALG[ANSIX9.31] Key(gen) (MOD: 1024, 1536, 2048, 3072, 4096; PubKey Values: 65537).

[NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) accepts Microsoft’s vendor-affirmed prerequisite NIST SP 800-90 RNG implementation that underlies the rDSA asymmetric cryptographic key generation as the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation.

### ECDSA algorithm

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received the following ECDSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 83 for
  + PKG: CURVES(P-256, P-384, P-512);
* Cert # 82 for
  + PKG: CURVES(P-256, P-384, P-512).

[NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) accepts Microsoft’s vendor-affirmed prerequisite NIST SP 800-90 RNG implementation that underlies the ECDSA asymmetric cryptographic key generation as the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has received the following ECDSA algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html):

* Cert # 83 for
  + PKG: CURVES(P-256, P-384, P-512);
* Cert # 82 for
  + PKG: CURVES(P-256, P-384, P-512).

[NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) accepts Microsoft’s vendor-affirmed prerequisite NIST SP 800-90 RNG implementation that underlies the ECDSA asymmetric cryptographic key generation as the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation.

### EC Diffie-Hellman Elliptic Curve-based key agreement algorithm

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) supports the key pair generation of the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm for key agreement with the following NIST P curves and corresponding key sizes:

* P-256 and 256 bits key size;
* P-384 and 384 bits key size;
* P-512 and 512 bits key size.

through its [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) and [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) interfaces.

Note that, currently, [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct validation tests for implementations of the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm for key agreement.

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation that underlies the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm asymmetric cryptographic key generation.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) supports the key pair generation of the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm for key agreement with the following NIST P curves and corresponding key sizes:

* P-256 and 256 bits key size;
* P-384 and 384 bits key size;
* P-512 and 512 bits key size.

through its [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) and [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) interfaces.

Note that, currently, [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) does not conduct validation tests for implementations of the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm for key agreement.

The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for its NIST SP 800-90 AES CTR\_DRBG algorithm implementation that underlies the EC Diffie-Hellman Elliptic Curve-based key agreement algorithm asymmetric cryptographic key generation.

Except the fact that neither the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) nor the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has yet received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for their implementations of the NIST SP800-90 AES CTR\_DRBG algorithm that underlies the following asymmetric key algorithms, this Commercial Grade OS Requirement Set “4.4.1.3” requirement is otherwise met.

* The rDSA algorithm;
* The ECDSA algorithm;
* The Diffie-Hellman Finite Field-based key agreement algorithm;
* The EC Diffie-Hellman Elliptic Curve-based key agreement algorithm.

## Addressing 4.4.1.4 “The OS crypto module shall perform key entry and output in accordance with the Level 1 Key Entry and Output requirements in [FIPS 140-2], “Security Requirements for Cryptographic Modules””

This requirement is addressed by the Windows OS as follows.

According to FIPS 140-2, the “Key Entry and Output” validation subarea is a subset of the “Cryptographic Key Management” validation area. In the case of the “Cryptographic Key Management” validation area, its assertions as specified in the [[Derived Test Requirements for FIPS 140-2, Security Requirements for Cryptographic Modules](http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/fips1402DTR.pdf)] for Security Level 1 and for Security Level 2 are identical. Therefore, the FIPS 140-2 overall Security Level 1 statuses for the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) suffice for the “Cryptographic Key Management” validation area, with respect to Security Level 2 also, and hence the “Key Entry and Output” validation subarea.

Consequently, this Commercial Grade OS Requirement Set “4.4.1.4” requirement is met.

## Addressing 4.4.1.5 “The OS crypto module shall provide the ability to protect secret keys, private keys, and critical cryptographic security parameters from unauthorized disclosure, modification, and substitution”

This requirement is addressed by the Windows OS as follows.

### Key isolation service

The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) is hosted in the LSASS.exe Windows OS process (i.e. the same process that also hosts the Windows OS Authentication Service). The service provides cryptographic key process isolation to cryptographic public/private key pairs and associated cryptographic operations for opening, creation, deletion, import, export, encrypting, decrypting, hash signing, signature verification, and secret agreement.

The ncrypt.dll is the wrapper for the FIPS 140-2 validated [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to support the long lived private key storage and retrieval in the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) when using the public/private key cryptographic services of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll).

The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) loads the ncrypt.dll during the initialization for its local RPC interfaces, which provide the key storage and retrieval services ([NCRYPT\_KEY\_STORAGE\_INTERFACE](http://msdn.microsoft.com/en-us/library/aa376201(VS.85).aspx) from the “Microsoft Software Key Storage Provider”). These local RPC interfaces are accessible through the following APIs, which are callable by any local authenticated user mode client subject.

* [NCryptOpenStorageProvider()](http://msdn.microsoft.com/en-us/library/aa376286(VS.85).aspx)
  + The default provider is the “Microsoft Software Key Storage Provider”.
  + The other built-in provider is the “Microsoft Smart Card Key Storage Provider”.
* [NCryptOpenKey()](http://msdn.microsoft.com/en-us/library/aa376284(VS.85).aspx)
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to open and read the key file containing the specified long lived private key (in the Windows OS [DPAPI](http://msdn.microsoft.com/en-us/library/ms995355.aspx) encrypted form) in the security context of the client subject from the “AppData\Roaming\Microsoft\Crypto\Keys” subdirectory under the root directory of the user (roaming) profile for the client subject.
  + The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5058](http://support.microsoft.com/kb/947226) “Key file operation (to) Read persisted key from file” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_FILE\_OPERATION.Id/ SE\_CNG\_ADT\_FILE\_READ) security audit record for success or failure after attempting to read the key file containing the specified long lived private key. The following informational items, when available, are indicated in the security audit record:
    - Security ID:
    - Account Name:
    - Account Domain:
    - Logon ID:
    - Provider Name:
    - Algorithm Name:
    - Key Name:
    - Key Type:
    - File Path:
    - Operation:
    - Return Code:.
  + After reading the key file content successfully, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) handler attempts to use the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) to obtain the clear text values of the protected properties associated with the specified long lived private key. The protected properties are:
    - [NCRYPT\_EXPORT\_POLICY\_PROPERTY (“Export Policy”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx);
    - [NCRYPT\_KEY\_USAGE\_PROPERTY (“Key Usage”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx);
    - MSCRYPT\_UI\_FLAG\_PROPERTY (“MSSP/UI Internal Flags”), indicating
      * MSCRYPT\_UI\_PASSWORD\_PROTECT\_FLAG;
    - [NCRYPT\_UI\_POLICY\_PROPERTY (“UI Policy”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx).
  + The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Open Key” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_OPEN\_KEY\_OPERATION) security audit record for success or failure after attempting to use the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) to obtain the clear text values of the protected properties associated with the specified long lived private key. The following informational items, when available, are indicated in the security audit record:
    - Security ID:
    - Account Name:
    - Account Domain:
    - Logon ID:
    - Provider Name:
    - Algorithm Name:
    - Key Name:
    - Key Type:
    - Operation:
    - Return Code:.
  + Upon obtaining the clear text values of the protected properties successfully, these properties could be made available to an authorized subject through the [NCryptGetProperty()](http://msdn.microsoft.com/en-us/library/aa375464(VS.85).aspx) API later on.
  + In the case, where the [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Open Key” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id) security audit record for success is generated, the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) returns a key handle for referencing the opened long lived private key as specified. The opened long lived private key remains inside the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx). It is still not in the clear text form until it is actually used during a subsequent request for a cryptographic operation associated with the key handle in [NCryptDecrypt()](http://msdn.microsoft.com/en-us/library/aa376249(VS.85).aspx), [NCryptExportKey()](http://msdn.microsoft.com/en-us/library/aa376263(VS.85).aspx), [NCryptSecretAgreement()](http://msdn.microsoft.com/en-us/library/aa376289(VS.85).aspx), or [NCryptSignHash()](http://msdn.microsoft.com/en-us/library/aa376295(VS.85).aspx).
* [NCryptCreatePersistedKey()](http://msdn.microsoft.com/en-us/library/aa376247(VS.85).aspx)
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to use [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) from the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to create an empty public/private key pair object through the “Microsoft Software Key Storage Provider”.
  + In the case, where the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) [BCryptGenerateKeyPair()](http://msdn.microsoft.com/en-us/library/aa375451(VS.85).aspx) successes, the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) returns a key handle for referencing the public/private key pair being created. Subsequently, the key handle can be used in [NCryptSetProperty()](http://msdn.microsoft.com/en-us/library/aa375504(VS.85).aspx) for setting specific properties associated with the referenced public/private key pair. After calling [NCryptFinalizeKey()](http://msdn.microsoft.com/en-us/library/aa376265(VS.85).aspx) for the key handle successfully, the referenced public/private key pair can be used during a subsequent request for a cryptographic operation associated with the key handle. The private key remains inside the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx). The public/private key pair may be exported in a subsequent use of [NCryptExportKey()](http://msdn.microsoft.com/en-us/library/aa376263(VS.85).aspx), subject to the export policy value in the [NCRYPT\_EXPORT\_POLICY\_PROPERTY (“Export Policy”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx) protected property of the private key.
* [NCryptGetProperty()](http://msdn.microsoft.com/en-us/library/aa375464(VS.85).aspx) about the “Microsoft Software Key Storage Provider”.
* [NCryptGetProperty()](http://msdn.microsoft.com/en-us/library/aa375464(VS.85).aspx) about the long lived key being created or opened.
* [NCryptSetProperty()](http://msdn.microsoft.com/en-us/library/aa375504(VS.85).aspx) for the “Microsoft Software Key Storage Provider”.
* [NCryptSetProperty()](http://msdn.microsoft.com/en-us/library/aa375504(VS.85).aspx) for the long lived key being created, in [NCryptCreatePersistedKey()](http://msdn.microsoft.com/en-us/library/aa376247(VS.85).aspx), through the referenced key handle
  + In the case, where the [NCryptSetProperty()](http://msdn.microsoft.com/en-us/library/aa375504(VS.85).aspx) request is used to import public and private key pair materials (in the clear text form) for the long lived public/private key pair being created, in [NCryptCreatePersistedKey()](http://msdn.microsoft.com/en-us/library/aa376247(VS.85).aspx), through the referenced key handle, the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) attempts to associate the imported public or private key pair materials (in the clear text form) with the long lived public/private key pair being created through the referenced key handle.
  + The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5059](http://support.microsoft.com/kb/947226) “Key migration operation (for) Import of persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_MIGRATION.Id/ SE\_CNG\_ADT\_KEY\_IMPORT) security audit record for success or failure after the imported public and private key pair material association attempt. The following informational items, when available, are indicated in the security audit record:
    - Security ID:
    - Account Name:
    - Account Domain:
    - Logon ID:
    - Provider Name:
    - Algorithm Name:
    - Key Name:
    - Key Type:
    - Operation:
    - Return Code:.
  + Note that the long lived key referenced by the key handle is not yet ready for use in a subsequent request for a cryptographic operation until a calling to [NCryptFinalizeKey()](http://msdn.microsoft.com/en-us/library/aa376265(VS.85).aspx) for the key handle has been successful.
* [NCryptImportKey()](http://msdn.microsoft.com/en-us/library/aa376276(VS.85).aspx) for importing a public key to receive the corresponding key handle to reference the imported public key
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to initialize a new key handle for referencing the imported public key so that the public key can be used in a subsequent request for a cryptographic operation.
  + The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5059](http://support.microsoft.com/kb/947226) “Key migration operation (for) Import of persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_MIGRATION.Id/ SE\_CNG\_ADT\_KEY\_IMPORT) security audit record for success or failure after the new key handle initialization attempt.
* [NCryptImportKey()](http://msdn.microsoft.com/en-us/library/aa376276(VS.85).aspx) for importing a public and private key pair blob in the PKCS # 7, PKCS # 8, or the standard CNG format to receive the corresponding key handle to reference the imported public/private key pair
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to initialize a new key handle for referencing the imported public and private key pair so that the public and private key pair can be used in a subsequent request for a cryptographic operation.
  + In the case where the [NCRYPT\_DO\_NOT\_FINALIZE\_FLAG](http://msdn.microsoft.com/en-us/library/aa376276(VS.85).aspx) is set in the request, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5059](http://support.microsoft.com/kb/947226) “Key migration operation (for) Import of persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_MIGRATION.Id/ SE\_CNG\_ADT\_KEY\_IMPORT) security audit record for success or failure after the new key handle initialization attempt. Note that the imported key pair referenced by the key handle is not yet ready for use in a subsequent request for a cryptographic operation until a calling to [NCryptFinalizeKey()](http://msdn.microsoft.com/en-us/library/aa376265(VS.85).aspx) for the key handle has been successful.
  + In the case where the [NCRYPT\_DO\_NOT\_FINALIZE\_FLAG](http://msdn.microsoft.com/en-us/library/aa376276(VS.85).aspx) is not set in the request,
    - the handler attempts to use [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) from the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to complete the import of the public/private key pair;
    - if the public/private key pair importation fails, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Create Key” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_CREATE\_KEY\_OPERATION) security audit record for failure; the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) also generates an [Event ID 5059](http://support.microsoft.com/kb/947226) “Key migration operation (for) Import of persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_MIGRATION.Id/ SE\_CNG\_ADT\_KEY\_IMPORT) security audit record for failure;
    - if the public/private key pair importation successes, the handler attempts to write the imported public/private key pair (in the Windows OS [DPAPI](http://msdn.microsoft.com/en-us/library/ms995355.aspx) encrypted form) and the values (in the Windows OS [DPAPI](http://msdn.microsoft.com/en-us/library/ms995355.aspx) encrypted form) of the associated protected properties to the key file of the associated key name in the security context of the client subject, where the key file resides in the “AppData\Roaming\Microsoft\Crypto\Keys” subdirectory under the root directory of the user (roaming) profile for the client subject;
    - the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5058](http://support.microsoft.com/kb/947226) “Key file operation (to) Write persisted key to file” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_FILE\_OPERATION.Id/ SE\_CNG\_ADT\_FILE\_WRITE) security audit record for success or failure after attempting to write to the key file;
    - if the key file writing fails, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) also generates an [Event ID 5059](http://support.microsoft.com/kb/947226) “Key migration operation (for) Import of persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_MIGRATION.Id/ SE\_CNG\_ADT\_KEY\_IMPORT) security audit record for failure;
    - if the key file writing is successful, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Create Key” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_CREATE\_KEY\_OPERATION) security audit record for success, and the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) also generates an [Event ID 5059](http://support.microsoft.com/kb/947226) “Key migration operation (for) Import of persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_MIGRATION.Id/ SE\_CNG\_ADT\_KEY\_IMPORT) security audit record for success.
* [NCryptFinalizeKey()](http://msdn.microsoft.com/en-us/library/aa376265(VS.85).aspx)
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to use [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) from the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to generate the public/private key pair or to complete the import of the public/private key pair for the referenced key handle based on the properties configured for the referenced key handle through a previous calling of [NCryptSetProperty()](http://msdn.microsoft.com/en-us/library/aa375504(VS.85).aspx).
  + In the case, where the public/private key pair generation or importation fails, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Create Key” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_CREATE\_KEY\_OPERATION) security audit record for failure.
  + In the case, where the public/private key pair generation or importation successes, the handler attempts to write the generated or imported public/private key pair (in the Windows OS [DPAPI](http://msdn.microsoft.com/en-us/library/ms995355.aspx) encrypted form) and the values (in the Windows OS [DPAPI](http://msdn.microsoft.com/en-us/library/ms995355.aspx) encrypted form) of the associated protected properties to the key file of the associated key name in the security context of the client subject. The key file resides in the “AppData\Roaming\Microsoft\Crypto\Keys” subdirectory under the root directory of the user (roaming) profile for the client subject.
  + The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5058](http://support.microsoft.com/kb/947226) “Key file operation (to) Write persisted key to file” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_FILE\_OPERATION.Id/ SE\_CNG\_ADT\_FILE\_WRITE) security audit record for success or failure after attempting to write to the key file.
  + In the case, where the key file writing is successful, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) also generates an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Create Key” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_CREATE\_KEY\_OPERATION) security audit record for success.
  + If a security descriptor (NCRYPT\_SECURITY\_DESCR\_PROPERTY) has been configured in a previous calling of [NCryptSetProperty()](http://msdn.microsoft.com/en-us/library/aa375504(VS.85).aspx) for the referenced key handle, then the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) assigns the security descriptor to the written key file for controlling the subsequent accesses to the key file.
* [NCryptSetProperty()](http://msdn.microsoft.com/en-us/library/aa375504(VS.85).aspx) for the created or opened public/private key pair associated with the referenced key handle
  + In the case, where the public/private key pair associated with the referenced key handle either
    - has already existed in a key file and is being opened;
    - has already been created and generated in [NCryptFinalizeKey()](http://msdn.microsoft.com/en-us/library/aa376265(VS.85).aspx);
    - has already been imported through [NCryptImportKey()](http://msdn.microsoft.com/en-us/library/aa376276(VS.85).aspx) or through [NCryptSetProperty()](http://msdn.microsoft.com/en-us/library/aa375504(VS.85).aspx), and been finalized in [NCryptFinalizeKey()](http://msdn.microsoft.com/en-us/library/aa376265(VS.85).aspx),

the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) does not allow the re-configuring of the following properties.

* + - the [NCRYPT\_EXPORT\_POLICY\_PROPERTY (“Export Policy”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx) protected property;
    - the [NCRYPT\_KEY\_USAGE\_PROPERTY (“Key Usage”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx) protected property;
    - the [NCRYPT\_UI\_POLICY\_PROPERTY (“UI Policy”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx) protected property;
    - the MSCRYPT\_UI\_PASSWORD\_PROTECT\_FLAG in the MSCRYPT\_UI\_FLAG\_PROPERTY (“MSSP/UI Internal Flags”) protected property;
    - the [NCRYPT\_LENGTH\_PROPERTY (“Length”)](http://msdn.microsoft.com/en-us/library/aa376242.aspx) property.
* [NCryptDeleteKey()](http://msdn.microsoft.com/en-us/library/aa376251(VS.85).aspx)
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to overwrite the content of the key file associated with the referenced key handle with zeros and to delete the key file from the “AppData\Roaming\Microsoft\Crypto\Keys” subdirectory under the root directory of the user (roaming) profile for the client subject.
  + The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5058](http://support.microsoft.com/kb/947226) “Key file operation (to) Delete key file” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_FILE\_OPERATION.Id/ SE\_CNG\_ADT\_FILE\_DELETE) security audit record for success or failure after attempting to overwrite and delete the key file.
* [NCryptFreeObject()](http://msdn.microsoft.com/en-us/library/aa376269(VS.85).aspx), where the object being freed is the private key associated with the referenced key handle
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to use [BCryptDestroyKey()](http://msdn.microsoft.com/en-us/library/aa375404(vs.85).aspx) from the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to destroy any private and public key associated with the referenced key handle from the memory of the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll).
  + If [BCryptDestroyKey()](http://msdn.microsoft.com/en-us/library/aa375404(vs.85).aspx) fails, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Delete Key” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_DELETE\_KEY\_OPERATION) security audit record for failure.
  + In the case, where [BCryptDestroyKey()](http://msdn.microsoft.com/en-us/library/aa375404(vs.85).aspx) successes to destroy all private and public keys associated with the referenced key handle, there is no need to generate an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Delete Key” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_DELETE\_KEY\_OPERATION) security audit record for success.
* [NCryptFreeBuffer()](http://msdn.microsoft.com/en-us/library/aa376267(VS.85).aspx).
* [NCryptEncrypt()](http://msdn.microsoft.com/en-us/library/aa376255(VS.85).aspx)
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to use [BCryptEncrypt()](http://msdn.microsoft.com/en-us/library/aa375421(VS.85).aspx) from the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to encrypt the specified data buffer with the public key associated with the referenced key handle.
  + If [BCryptEncrypt()](http://msdn.microsoft.com/en-us/library/aa375421(VS.85).aspx) fails, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Encrypt” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_ENCRYPT\_OPERATION) security audit record for failure.
  + In the case, where [BCryptEncrypt()](http://msdn.microsoft.com/en-us/library/aa375421(VS.85).aspx) successes to encrypt the specified data buffer, there is no need to generate an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Encrypt” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_ENCRYPT\_OPERATION) security audit record for success.
* [NCryptDecrypt()](http://msdn.microsoft.com/en-us/library/aa376249(VS.85).aspx)
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to use [BCryptDecrypt()](http://msdn.microsoft.com/en-us/library/aa375391(VS.85).aspx) from the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to decrypt the specified data buffer with the private key associated with the referenced key handle.
  + If the private key associated with the referenced key handle is still in the Windows OS [DPAPI](http://msdn.microsoft.com/en-us/library/ms995355.aspx) encrypted form after being read from its key file, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) handler needs to attempt to use the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) to obtain the private key in the clear text before using it in [BCryptDecrypt()](http://msdn.microsoft.com/en-us/library/aa375391(VS.85).aspx).
  + If the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) fails to obtain the private key in the clear text, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5060](http://support.microsoft.com/kb/947226) “Verification operation failed (in the case of) Failed to unprotect persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_VERIFICATION\_FAILURE.Id/SE\_CNG\_ADT\_UNPROTECT\_FAILURE) security audit record for failure. The following informational items, when available, are indicated in the security audit record:
    - Security ID:
    - Account Name:
    - Account Domain:
    - Logon ID:
    - Provider Name:
    - Algorithm Name:
    - Key Name:
    - Key Type:
    - Return Code:.
  + In the case, where the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) successes to obtain the private key in the clear text, there is no need to generate a security audit record for success.
  + If [BCryptDecrypt()](http://msdn.microsoft.com/en-us/library/aa375391(VS.85).aspx) fails, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Decrypt” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_DECRYPT\_OPERATION) security audit record for failure.
  + In the case, where [BCryptDecrypt()](http://msdn.microsoft.com/en-us/library/aa375391(VS.85).aspx) successes to decrypt the specified data buffer, there is no need to generate an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Decrypt” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_DECRYPT\_OPERATION) security audit record for success.
* [NCryptSignHash()](http://msdn.microsoft.com/en-us/library/aa376295(VS.85).aspx)
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to use [BCryptSignHash()](http://msdn.microsoft.com/en-us/library/aa375510(VS.85).aspx) from the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to sign the specified hash data buffer with the private key associated with the referenced key handle.
  + If the private key associated with the referenced key handle is still in the Windows OS [DPAPI](http://msdn.microsoft.com/en-us/library/ms995355.aspx) encrypted form after being read from its key file, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) handler needs to attempt to use the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) to obtain the private key in the clear text before using it in [BCryptSignHash()](http://msdn.microsoft.com/en-us/library/aa375510(VS.85).aspx).
  + If the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) fails to obtain the private key in the clear text, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5060](http://support.microsoft.com/kb/947226) “Verification operation failed (in the case of) Failed to unprotect persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_VERIFICATION\_FAILURE.Id/SE\_CNG\_ADT\_UNPROTECT\_FAILURE) security audit record for failure.
  + In the case, where the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) successes to obtain the private key in the clear text, there is no need to generate a security audit record for success.
  + If [BCryptSignHash()](http://msdn.microsoft.com/en-us/library/aa375510(VS.85).aspx) fails, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Sign hash” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_SIGN\_HASH\_OPERATION) security audit record for failure.
  + In the case, where [BCryptSignHash()](http://msdn.microsoft.com/en-us/library/aa375510(VS.85).aspx) successes to sign the specified hash data buffer, there is no need to generate an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Sign hash” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_SIGN\_HASH\_OPERATION) security audit record for success.
* [NCryptVerifySignature()](http://msdn.microsoft.com/en-us/library/aa376298(VS.85).aspx)
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to use [BCryptVerifySignature()](http://msdn.microsoft.com/en-us/library/aa375515(VS.85).aspx) from the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to verify that the specified signature value matches the specified hash value, with the public key associated with the referenced key handle.
  + If [BCryptVerifySignature()](http://msdn.microsoft.com/en-us/library/aa375515(VS.85).aspx) fails, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5060](http://support.microsoft.com/kb/947226) “Verification operation failed (in the case of) Signature verification failed” (SE\_AUDITID\_ETW\_NCRYPT\_VERIFICATION\_FAILURE.Id/SE\_CNG\_ADT\_SIGNATURE\_VERIFY\_FAILURE) security audit record for failure.
  + In the case, where [BCryptVerifySignature()](http://msdn.microsoft.com/en-us/library/aa375515(VS.85).aspx) successes to verify that the specified signature value matches the specified hash value, there is no need to generate a security audit record for success.
* [NCryptSecretAgreement()](http://msdn.microsoft.com/en-us/library/aa376289(VS.85).aspx)
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to use [BCryptSecretAgreement()](http://msdn.microsoft.com/en-us/library/aa375496(VS.85).aspx) from the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to calculate a secret agreement value from the private key associated with the referenced (private) key handle and the public key associated with the referenced (public) key handle. The calculated secret agreement value stays inside the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx). Only the agreement secret handle is returned to the requester subject for referencing the secret agreement value in a subsequent [NCryptDeriveKey()](http://msdn.microsoft.com/en-us/library/aa376252(VS.85).aspx) request.
  + If the private key associated with the referenced (private) key handle is still in the Windows OS [DPAPI](http://msdn.microsoft.com/en-us/library/ms995355.aspx) encrypted form after being read from its key file, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) handler needs to attempt to use the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) to obtain the private key in the clear text before using it in [BCryptSecretAgreement()](http://msdn.microsoft.com/en-us/library/aa375496(VS.85).aspx).
  + If the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) fails to obtain the private key in the clear text, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5060](http://support.microsoft.com/kb/947226) “Verification operation failed (in the case of) Failed to unprotect persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_VERIFICATION\_FAILURE.Id/SE\_CNG\_ADT\_UNPROTECT\_FAILURE) security audit record for failure.
  + In the case, where the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) successes to obtain the private key in the clear text, there is no need to generate a security audit record for success.
  + If [BCryptSecretAgreement()](http://msdn.microsoft.com/en-us/library/aa375496(VS.85).aspx) fails, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (for) Secret agreement” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_SECRET\_AGREEMENT\_OPERATION) security audit record for failure.
  + In the case, where [BCryptSecretAgreement()](http://msdn.microsoft.com/en-us/library/aa375496(VS.85).aspx) successes to calculate a secret agreement value, there is no need to generate an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (for) Secret agreement” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_SECRET\_AGREEMENT\_OPERATION) security audit record for success.
* [NCryptDeriveKey()](http://msdn.microsoft.com/en-us/library/aa376252(VS.85).aspx)
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler attempts to use [BCryptDeriveKey()](http://msdn.microsoft.com/en-us/library/aa375393(VS.85).aspx) from the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to derive a session key, for output to the requester subject, from the secret agreement value associated with the referenced agreement secret handle, according to the specified key derivation function identifier.
  + There is no need to generate a security audit record for the success or failure outcome of the session key derivation from the secret agreement value.
* [NCryptIsAlgSupported()](http://msdn.microsoft.com/en-us/library/aa376278(VS.85).aspx).
* [NCryptEnumAlgorithms()](http://msdn.microsoft.com/en-us/library/aa376257(VS.85).aspx).
* [NCryptEnumKeys()](http://msdn.microsoft.com/en-us/library/aa376259(VS.85).aspx).
* [NCryptExportKey()](http://msdn.microsoft.com/en-us/library/aa376263(VS.85).aspx)
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), subject to the export policy value in the [NCRYPT\_EXPORT\_POLICY\_PROPERTY (“Export Policy”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx) protected property of the private key associated with the referenced key handle, the handler attempts to use [BCryptExportKey()](http://msdn.microsoft.com/en-us/library/aa375434(VS.85).aspx) from the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to export the public and private key pair associated with the referenced key handle, in the specified formats.
  + If the private key associated with the referenced key handle is still in the Windows OS [DPAPI](http://msdn.microsoft.com/en-us/library/ms995355.aspx) encrypted form after being read from its key file, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) handler needs to attempt to use the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) to obtain the private key in the clear text before using it in [BCryptExportKey()](http://msdn.microsoft.com/en-us/library/aa375434(VS.85).aspx).
  + If the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) fails to obtain the private key in the clear text, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5060](http://support.microsoft.com/kb/947226) “Verification operation failed (in the case of) Failed to unprotect persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_VERIFICATION\_FAILURE.Id/SE\_CNG\_ADT\_UNPROTECT\_FAILURE) security audit record for failure. The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) also generates an [Event ID 5059](http://support.microsoft.com/kb/947226) “Key migration operation (for) Export of persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_MIGRATION.Id/SE\_CNG\_ADT\_KEY\_EXPORT) security audit record for failure.
  + In the case, where the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) successes to obtain the private key in the clear text, there is no need to generate a security audit record for success.
  + Finally, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5059](http://support.microsoft.com/kb/947226) “Key migration operation (for) Export of persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_MIGRATION.Id/SE\_CNG\_ADT\_KEY\_EXPORT) security audit record for success or failure after attempting to use [BCryptExportKey()](http://msdn.microsoft.com/en-us/library/aa375434(VS.85).aspx) for obtaining the public and private key pair according to the specified formats.
* [NCryptNotifyChangeKey()](http://msdn.microsoft.com/en-us/library/aa376281(VS.85).aspx).
* [NCryptFreeObject()](http://msdn.microsoft.com/en-us/library/aa376269(VS.85).aspx), where the object being freed is an agreement secret object which has been calculated from a previous calling of [NCryptSecretAgreement()](http://msdn.microsoft.com/en-us/library/aa376289(VS.85).aspx).
  + After the request reaches the handler within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll), the handler uses [BCryptDestroySecret()](http://msdn.microsoft.com/en-us/library/aa375407(VS.85).aspx) from the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to destroy the agreement secret value associated with the referenced key handle from the memory of the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll).
  + There is no need to generate a security audit record for the success or failure outcome of the destroying of the agreement secret value.

### Protections provided by the Windows OS key isolation service

As mentioned in the above, every long lived private key, belonging to a user account, resides in its specific key file. While residing in the key file, the private key is protected by the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx). The specific key file resides in the “AppData\Roaming\Microsoft\Crypto\Keys” subdirectory under the root directory of the user (roaming) profile for the user account. Because the key file is a part of the user (roaming) profile for the user account, it roams from the Active Directory [user profile](http://technet2.microsoft.com/windowsserver/en/library/093238f3-5064-470e-a281-0eb1c28b9cf01033.mspx?mfr=true) path location, for the user account, to any individual Windows OS machines where the user account is allowed to be logged onto by a user. The protection provided by the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) ensures that the long lived private key remains encrypted until it is explicitly unprotected with the DPAPI [CryptUnprotectData()](http://msdn.microsoft.com/en-us/library/aa380882(VS.85).aspx) interface. The security of the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) protection is described in the following two sections of this paper.

* “Kerberos security provider notifies the local Windows OS data protection API manger”;
* “NTLM security provider notifies the local Windows OS data protection API manger”.

This protection from the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) ensures that only a user either

* successfully logging on to the private key owner user account;
* physically possessing the private key owner user account’s password key disk according to the user manual of the “[Forgotten Password Wizard](http://support.microsoft.com/kb/930381)”,

can gain access to the clear text of the private key.

As well as the encryption with the user account’s master key encryption key for the private key, the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) protection for the private key also includes a HMAC check to ensure the private key’s integrity.

Additionally, if the NCRYPT\_UI\_FORCE\_HIGH\_PROTECTION\_FLAG is set in the [NCRYPT\_UI\_POLICY\_PROPERTY (“UI Policy”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx) protected property of the private key, then the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) [CryptUnprotectData()](http://msdn.microsoft.com/en-us/library/aa380882(VS.85).aspx) interface requires the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) to collect an additional password from the private key owner subject for the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) to obtain the clear text private key. This behavior can be configured by an authorized administrator via the following registry key in the “[Force strong key protection for user keys stored on the computer](http://technet.microsoft.com/en-us/library/cc738035.aspx)” policy. The default option of the policy is the “User input is not required when new keys are stored and used” option (i.e. not to require an additional password).

* HKEY\_LOCAL\_MACHINE\Software\Policies\Microsoft\Cryptography\ForceKeyProtection.

While residing in the “AppData\Roaming\Microsoft\Crypto\Keys” subdirectory under the root directory of the user (roaming) profile for the user account, an individual key file is also protected by the associated security descriptor when an access to the key file is attempted. The default permission is to grant the full access to the private key owner user account, the local system and the “Administrators” group. A specific security descriptor can be configured by using the [NCRYPT\_SECURITY\_DESCR\_PROPERTY (“Security Descr”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx) property for the private key in [NCryptSetProperty()](http://msdn.microsoft.com/en-us/library/aa375504(VS.85).aspx).

For a public/private key pair that is created and generated within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx), the private key clear text appears inside the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) only.

For a public/private key pair that resides in a key file, the private key clear text also appears inside the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) only, after being read from the key file by the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) through the service of the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) protection. The [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) protection ensures that the user account of the subject, requesting the private key for use in a cryptographic operation through the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx), is indeed the owner user account of the requested private key.

If the “User must enter a password each time they use a key” option is configured in the “[Force strong key protection for user keys stored on the computer](http://technet.microsoft.com/en-us/library/cc738035.aspx)” policy, then the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) additionally requires the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) to collect an additional password from the private key owner subject.

For a (clear text) public/private key pair that is imported into the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx), the private key is encrypted through the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) protection before being written to the corresponding key file for persistent storage.

Subject to the export policy value in the [NCRYPT\_EXPORT\_POLICY\_PROPERTY (“Export Policy”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx) protected property of the private key, the clear text of the private key may be exported with its corresponding public key from the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) through [NCryptExportKey()](http://msdn.microsoft.com/en-us/library/aa376263(VS.85).aspx).

In the secret agreement cryptographic operation though [NCryptSecretAgreement()](http://msdn.microsoft.com/en-us/library/aa376289(VS.85).aspx), the calculated secret agreement value stays inside the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx). Only individual specific derived session keys from the calculated secret agreement value are allowed to leave the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) through [NCryptDeriveKey()](http://msdn.microsoft.com/en-us/library/aa376252(VS.85).aspx).

The FIPS-140-2 validated [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) takes additional care when it is requested to destroy specific private keys and secret agreement values from the memory.

When [BCryptDestroyKey()](http://msdn.microsoft.com/en-us/library/aa375404(vs.85).aspx) of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) is used to destroy a clear text private key currently existing in the memory, [BCryptDestroyKey()](http://msdn.microsoft.com/en-us/library/aa375404(vs.85).aspx) does not only overwrite the memory buffer holding the private key with zeros as explained in the “Addressing 5.1.1.2 “Objects associated with cryptographic keys and critical cryptographic security parameters shall be destroyed as described in the Commercial Grade OS Requirement Set “4.4.1.8” requirement”” section of paper, [BCryptDestroyKey()](http://msdn.microsoft.com/en-us/library/aa375404(vs.85).aspx) also verifies that every slot of the overwritten memory buffer is indeed zero. If [BCryptDestroyKey()](http://msdn.microsoft.com/en-us/library/aa375404(vs.85).aspx) is used within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx), a failure of this read-zero-verify action further results in the generation of an [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Failed to zero secret data” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_ZERO\_DATA\_FAILURE) security audit record for failure. The following informational items, when available, are indicated in the security audit record:

* Security ID:
* Account Name:
* Account Domain:
* Logon ID:
* Provider Name:
* Algorithm Name:
* Return Code:.

Similarly, when [BCryptDestroySecret()](http://msdn.microsoft.com/en-us/library/aa375407(VS.85).aspx)of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) is used to destroy a calculated clear text secret agreement value currently existing in the memory, [BCryptDestroySecret()](http://msdn.microsoft.com/en-us/library/aa375407(VS.85).aspx) does not only overwrite the memory buffer holding the calculated secret agreement value with zeros as explained in the “Addressing 5.1.1.2 “Objects associated with cryptographic keys and critical cryptographic security parameters shall be destroyed as described in the Commercial Grade OS Requirement Set “4.4.1.8” requirement”” section of paper, [BCryptDestroySecret()](http://msdn.microsoft.com/en-us/library/aa375407(VS.85).aspx) also verifies that every slot of the overwritten memory buffer is indeed zero. If [BCryptDestroySecret()](http://msdn.microsoft.com/en-us/library/aa375407(VS.85).aspx) is used within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx), a failure of this read-zero-verify action further results in the generation of an [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Failed to zero secret data” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_ZERO\_DATA\_FAILURE) security audit record for failure.

In summary, the most important aspect of the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) protection is architectural. The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) exists in the LSASS.exe Windows OS process (i.e. the same process that also hosts the Windows OS Authentication Service) for handling the cryptographic operations associated with a long lived public/private key pair. Together with the security of the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx), the special care taken when a private key is destroyed, and the availability of auditing, the overall [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) protection provides the ability to protect the long lived private keys and the calculated secret agreement value, from unauthorized disclosure, modification, and substitution.

Consequently, this Commercial Grade OS Requirement Set “4.4.1.5” requirement is met.

## Addressing 4.4.1.6 “The OS crypto module shall provide the ability to protect public keys from unauthorized modification, and substitution”

This requirement is addressed by the Windows OS as follows.

In general, public keys are associated with their public key certificates. The integrity of a public key is ensured when its public key certificate is verified using the Windows OS [WinVerifyTrustEx()](http://msdn.microsoft.com/en-us/library/aa388209(VS.85).aspx) function, through the verification of the corresponding public key certificate chain. Root public keys are also associated with root certificates. However, a public key certificate chain is not available to validate any root certificate, and hence its public key. When root certificates exist in the Windows OS “Root” certificate stores, the Windows OS needs to maintain their integrity. The Windows OS “Root” certificate stores reside in the registry. The actual registry locations of these Windows OS “Root” certificate stores are listed in the MSDN description of “[System Store Locations](http://msdn.microsoft.com/en-us/library/aa388136(VS.85).aspx)” for public key certificates.

When a subject belonging to a user account attempts to open a Windows OS “Root” certificate store through the Windows OS [CertOpenStore()](http://msdn.microsoft.com/en-us/library/aa376559(VS.85).aspx) function with [CERT\_STORE\_PROV\_SYSTEM\_REGISTRY\_W/CERT\_STORE\_PROV\_SYSTEM\_REGISTRY\_A](http://msdn.microsoft.com/en-us/library/aa377575(VS.85).aspx), the root certificates in the store are enumerated and they are compared against the list of protected roots for the user account. If an enumerated root certificate does not have a certificate SHA-1 hash (i.e. [CERT\_SHA1\_HASH\_PROP\_ID](http://msdn.microsoft.com/en-us/library/aa376079(VS.85).aspx) property) entry in the list of protected roots, the root certificate is deleted from the store so that it could not be used by the subject after opening the store. The SHA-1 hash of a public key certificate is the value of the [CERT\_SHA1\_HASH\_PROP\_ID](http://msdn.microsoft.com/en-us/library/aa376079(VS.85).aspx) property of the certificate as calculated by [CryptHashCertificate2()](http://msdn.microsoft.com/en-us/library/aa380201(VS.85).aspx).

The list of protected roots (consisting of a list of certificate SHA-1 hashes) for the user account resides in the “certificates” value of the following registry key for the user account. This list is managed locally by the Windows OS protected root service (residing in “cryptsvc.dll”).

* HKEY\_CURRENT\_USER\Software\Microsoft\SystemCertificates\Root\ProtectedRoots.

The security descriptor associated with this registry key provides the necessary protection for the registry key. By default, the security descriptor DACL grants only

* full access tothe Windows OS protected root service (i.e. cryptsvc as the service SID: S-1-5-80-242729624-280608522-2219052887-3187409060-2225943459);
* read only to the user account.

In addition, the security descriptor MACL assigns the MandatoryLevelHigh (3) integrity level to the registry key. As a result, only a subject labeled with the MandatoryLevelHigh (3) integrity level or high is able to modify the list of protected roots for a user account in the above “ProtectedRoots” registry key. The Protected Root Service runs in the security context of the local system and hence it is labeled the MandatoryLevelHigh (3) integrity level.

The Windows OS protected root service (i.e. cryptsvc.dll) accepts requests from a local subject belonging to a user account to make the following adding and deleting changes to the list of protected roots for user account. For a request to be successful, the subject must be labeled at least MandatoryLevelMedium (2).

* Adding a root certificate to the user account’s list of protected roots;
* Deleting a root certificate from the user account’s list of protected roots.

When a MandatoryLevelMedium (2) local subject of the user account requests the Windows OS protected root service to add a root certificate to the user account’s list of protected roots, the Windows OS protected root service displays a “Security Warning” dialog message box on the user interactive application display area (in the window terminal session where the user account is logged on) to ask the interactive user for his/her consent.

* “You are about to install a certificate from a certification authority (CA) claiming to represent: <a specific issuer name>”
* “Windows cannot validate that the certificate is actually from “<the specific issuer name>”. You should confirm its origin by contacting “<the specific issuer name>””.
* “The following number will assist you in this process: Thumbprint (sha1): <the certificate SHA-1 hash>”.
* “Warning: If you install this root certificate, Windows will automatically trust any certificate issued by this CA. Installing a certificate with an unconfirmed thumbprint is a security risk.”
* “If you click “Yes”, you acknowledge this risk. Do you want to install this certificate?”

The dialog message box also includes the following information items about the root certificate being requested for the addition:

* Issuer;
* The certificate SHA-1 hash (i.e. [CERT\_SHA1\_HASH\_PROP\_ID](http://msdn.microsoft.com/en-us/library/aa376079(VS.85).aspx) property).

If the interactive user clicks the “Yes” button, then the SHA-1 hash value of the root certificate is added to the user account’s list of protected roots. Otherwise, the request to the Windows OS protected root service for adding the root certificate is rejected.

When a MandatoryLevelMedium (2) local subject of the user account requests the Windows OS protected root service to delete a root certificate from the user account’s list of protected roots, the Windows OS protected root service displays a “Root Certificate Store” informational dialog message box on the user interactive application display area (in the window terminal session where the user account is logged on) to ask the interactive user for his/her consent.

* “Do you want to DELETE the following certificate from the Root Store?”

The dialog message box also includes the following information items about the root certificate being requested for the deletion:

* Root certificate subject name;
* Issuer: Self issued;
* Time Validity: ;
* Serial Number: ;
* Thumbprint (sha1): .

If the interactive user clicks the “Yes” button, then the SHA-1 hash value of the root certificate is deleted from the user account’s list of protected roots. Otherwise, the request to the Windows OS protected root service for deleting the root certificate is rejected.

Finally, to support public key certificate validation where the public key certificate chain ends in a Microsoft root public key as provided by the [CERT\_CHAIN\_POLICY\_MICROSOFT\_ROOT](http://msdn.microsoft.com/en-us/library/aa377163.aspx) policy, the public key ([CERT\_PUBLIC\_KEY\_INFO](http://msdn.microsoft.com/en-us/library/aa377463(VS.85).aspx)) SHA-1 hashes of the two Microsoft root public keys are necessarily embedded in [crypt32.dll](http://msdn.microsoft.com/en-us/library/aa379884(VS.85).aspx).

In a running system after the proper installation of Windows OS, the Windows OS system executable image files such as [crypt32.dll](http://msdn.microsoft.com/en-us/library/aa379884(VS.85).aspx) reside in the “$SystemRoot$\System32” directory of the local file system. These files have the default file object security descriptor, and the directory has the default file directory object security descriptor as follows.

* The default file object security descriptor has the Windows OS Trusted Installer (S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464) as the owner. It grants the built-in “Administrators” group, the local system, and the built-in “Users” group only the generic read and execute access rights.
* The default file directory object security descriptor has the Windows OS Trusted Installer (S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464) as the owner. It grants the built-in “Administrators” group and the local system only the modify access rights, and it grants the built-in “Users” group only the generic read and execute access rights.

Due to

* the security descriptor applied to the “ProtectedRoots” registry key for a user account:
  + including an explicit MandatoryLevelHigh (3) [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE for the write ([SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class;
  + including a restrictive DACL;
* the security descriptor applied to the [crypt32.dll](http://msdn.microsoft.com/en-us/library/aa379884(VS.85).aspx) image file which contains the public key ([CERT\_PUBLIC\_KEY\_INFO](http://msdn.microsoft.com/en-us/library/aa377463(VS.85).aspx)) SHA-1 hashes of the two Microsoft root public keys:
  + implying a default MandatoryLevelMedium (2) [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE for the write ([SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class;
  + including a restrictive DACL;
* the user consent on the user interactive application display area (in the window terminal session where the user account is logged on) when adding or deleting an element in the user account’s list of protected roots,

this Commercial Grade OS Requirement Set “4.4.1.6” requirement is met.

## Addressing 4.4.1.7 “The OS crypto module shall provide the ability to correctly associate stored cryptographic keys (secret, private, or public) with the entity (e.g. person, group, or subject) to which the key is assigned”

This requirement is addressed by the Windows OS as follows.

As mentioned in the “Protections provided by the Windows OS key isolation service” section of this paper, every long lived private key, belonging to a user account, resides in its specific key file. While residing in the key file, the private key is protected by the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx). The specific key file resides in the “AppData\Roaming\Microsoft\Crypto\Keys” subdirectory under the root directory of the user (roaming) profile for the user account. Because the key file is a part of the user (roaming) profile for the user account, it roams from the Active Directory [user profile](http://technet2.microsoft.com/windowsserver/en/library/093238f3-5064-470e-a281-0eb1c28b9cf01033.mspx?mfr=true) path location, for the user account, to any individual Windows OS machines where the user account is allowed to be logged onto by a user. The protection provided by the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) ensures that the long lived private key remains encrypted until it is explicitly unprotected with the DPAPI [CryptUnprotectData()](http://msdn.microsoft.com/en-us/library/aa380882(VS.85).aspx) interface. The security of the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) protection is described in the following two sections of this paper.

* “Kerberos security provider notifies the local Windows OS data protection API manger”;
* “NTLM security provider notifies the local Windows OS data protection API manger”.

This protection from the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) ensures that only a user either

* successfully logging on to the private key owner user account;
* physically possessing the private key owner user account’s password key disk according to the user manual of the “[Forgotten Password Wizard](http://support.microsoft.com/kb/930381)”,

can gain access to the clear text of the private key.

Additionally, if the NCRYPT\_UI\_FORCE\_HIGH\_PROTECTION\_FLAG is set in the [NCRYPT\_UI\_POLICY\_PROPERTY (“UI Policy”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx) protected property of the private key, then the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) [CryptUnprotectData()](http://msdn.microsoft.com/en-us/library/aa380882(VS.85).aspx) interface requires the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) to collect an additional password from the private key owner subject for the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) to obtain the clear text private key. This behavior can be configured by an authorized administrator via the following registry key in the “[Force strong key protection for user keys stored on the computer](http://technet.microsoft.com/en-us/library/cc738035.aspx)” policy. The default option of the policy is the “User input is not required when new keys are stored and used” option (i.e. not to require an additional password).

* HKEY\_LOCAL\_MACHINE\Software\Policies\Microsoft\Cryptography\ForceKeyProtection.

While residing in the “AppData\Roaming\Microsoft\Crypto\Keys” subdirectory under the root directory of the user (roaming) profile for the user account, an individual key file is also protected by the associated security descriptor when an access to the key file is attempted. The default permission is to grant the full access to the private key owner user account, the local system and the “Administrators” group. A specific security descriptor can be configured by using the [NCRYPT\_SECURITY\_DESCR\_PROPERTY (“Security Descr”)](http://msdn.microsoft.com/en-us/library/aa376242(VS.85).aspx) property for the private key in [NCryptSetProperty()](http://msdn.microsoft.com/en-us/library/aa375504(VS.85).aspx).

For a public key and its associated public key certificate, we recall that there is the below [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) attribute for a Windows OS domain based user account as explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement. This attribute provides a mechanism to establish the binding with the specific user account and the account’s public keys and associated public key certificates.

* [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) (if the user account is domain based)
  + It contains associated mappings for X.509 public key certificates or external Kerberos user accounts to the user account for the purpose of public key certificate or smart card based user authentication;
  + By default, a standard user does not possess the right to update this attribute value;

Consequently, this Commercial Grade OS Requirement Set “4.4.1.7” requirement is met.

## Addressing 4.4.1.8 “The OS crypto module shall destroy cryptographic keys in accordance with a specific cryptographic key zeroization method”

The Commercial Grade OS Requirement Set requires the specific cryptographic key zeroization method to meet the following requirements.

1. Key zeroization requirements in [FIPS PUB 140-2], “Security Requirements for Cryptographic Modules.”
2. Zeroization of all plaintext cryptographic keys and all other critical cryptographic security parameters shall be immediate and complete.
3. Each intermediate storage area for plaintext cryptographic key /critical cryptographic security parameter (i.e. any storage, such as memory buffers, that is included in the path of such data) shall be zeroized.
4. Zeroization shall be executed by one of the following two selections:
   1. For non-volatile memories other than EEPROM and Flash, the zeroization shall be executed by overwriting three or more times using a different alternating data pattern each time;
   2. For volatile memories and non-volatile EEPROM and Flash memories, the zeroization shall be executed by a single direct overwrite consisting of a pseudo random pattern, followed by a read-verify.

This requirement is addressed by the descriptions given in the justification text for addressing the Commercial Grade OS Requirement Set “5.1.1.2” requirement. The descriptions reside in the following section of this paper.

* “Addressing 5.1.1.2 “Objects associated with cryptographic keys and critical cryptographic security parameters shall be destroyed as described in the Commercial Grade OS Requirement Set “4.4.1.8” requirement””.

Consequently, this Commercial Grade OS Requirement Set “4.4.1.8” requirement is addressed.

# Meeting the “Cryptographic Support Cryptographic Key Management Management Requirements”

In the Commercial Grade OS Requirement Set, there is no individual management requirement under the heading of “Cryptographic Support Cryptographic Key Management Management Requirements”.

# Meeting the “Cryptographic Support Cryptographic Key Management Audit Requirements”

In the Commercial Grade OS Requirement Set, there are 2 individual audit requirements under the heading of “Cryptographic Support Cryptographic Key Management Audit Requirements”. They are listed as “4.4.3.n”, where n = 1 and 2.

## Addressing 4.4.3.1 “The OS shall provide the ability to audit attempts to violate the restrictions specified in “4.4.1.5””

This requirement is addressed by the Windows OS as follows.

As described in the justification text for addressing the Commercial Grade OS Requirement Set “4.4.1.5” requirement, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) is capable to generate the following security audit records.

* The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5058](http://support.microsoft.com/kb/947226) “Key file operation (to) Read persisted key from file” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_FILE\_OPERATION.Id/ SE\_CNG\_ADT\_FILE\_READ) security audit record for success or failure after attempting to read the key file containing the specified long lived private key.
* The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Open Key” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_OPEN\_KEY\_OPERATION) security audit record for success or failure after attempting to use the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) to obtain the clear text values of the protected properties associated with the specified long lived private key.
* If the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) fails to obtain the private key in the clear text, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5060](http://support.microsoft.com/kb/947226) “Verification operation failed (in the case of) Failed to unprotect persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_VERIFICATION\_FAILURE.Id/SE\_CNG\_ADT\_UNPROTECT\_FAILURE) security audit record for failure. There is no need to generate a security audit record for success in the case where the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) successes to obtain the private key in the clear text.
* The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5059](http://support.microsoft.com/kb/947226) “Key migration operation (for) Export of persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_MIGRATION.Id/SE\_CNG\_ADT\_KEY\_EXPORT) security audit record for success or failure after attempting to use [BCryptExportKey()](http://msdn.microsoft.com/en-us/library/aa375434(VS.85).aspx) for obtaining the public and private key pair according to the specified formats.
* The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5059](http://support.microsoft.com/kb/947226) “Key migration operation (for) Import of persistent cryptographic key” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_MIGRATION.Id/ SE\_CNG\_ADT\_KEY\_IMPORT) security audit record for success or failure after the imported public and private key pair material association attempt.
* The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5061](http://support.microsoft.com/kb/947226) “Cryptographic operation (to) Create Key” (SE\_AUDITID\_ETW\_NCRYPT\_OPERATION\_EVENT.Id/ SE\_CNG\_ADT\_CREATE\_KEY\_OPERATION) security audit record for success or failure.
* The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5058](http://support.microsoft.com/kb/947226) “Key file operation (to) Write persisted key to file” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_FILE\_OPERATION.Id/ SE\_CNG\_ADT\_FILE\_WRITE) security audit record for success or failure after attempting to write to the key file.
* The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5058](http://support.microsoft.com/kb/947226) “Key file operation (to) Delete key file” (SE\_AUDITID\_ETW\_NCRYPT\_KEY\_FILE\_OPERATION.Id/ SE\_CNG\_ADT\_FILE\_DELETE) security audit record for success or failure after attempting to overwrite and delete the key file.

Recall also that key files reside in the “AppData\Roaming\Microsoft\Crypto\Keys” subdirectory under the root directory of the user (roaming) profile for the client subject. As a named file object, key files are subject to the auditing due to the Discretionary Access Control policy decisions as shown in Table 1 of the “Addressing 2.1.1.1 “The OS shall enforce a Discretionary Access Control policy on all subjects, all named objects and all access operations among them”” section of paper. Therefore, as described in the justification text for addressing the Commercial Grade OS Requirement Set “2.1.3.1” requirement, the following security audit records are generated when the security descriptor associated with a key file includes a [system access control list (SACL)](http://msdn.microsoft.com/en-us/library/aa375723(VS.85).aspx) of the corresponding [SYSTEM\_AUDIT\_ACE](http://msdn.microsoft.com/en-us/library/aa379616(VS.85).aspx) ACEs.

* [Event ID 4656](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OPEN\_HANDLE\_value) “A handle to an object was requested” for success or failure.
* [Event ID 4663](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OBJECT\_ACCESS\_value) “An attempt was made to access an object” for success or failure.

Finally, when a key file (as a file object) is deleted successfully through the corresponding interface of the Windows OS IO manager, the Windows OS NTFS file system (ntfs.sys) also is capable to generate an [Event ID 4660](http://support.microsoft.com/kb/947226) “An object was deleted” (SE\_AUDITID\_ETW\_DELETE\_OBJECT\_value) security audit record for success with the following informational items when available:

* Security ID:
* Account Name:
* Account Domain:
* Logon ID:
* Object Server:
* Handle ID:
* Process ID:
* Process Name:
* Transaction ID:.

Consequently, this Commercial Grade OS Requirement Set “4.4.3.1” requirement is met.

## Addressing 4.4.3.2 “The OS shall provide the ability to audit attempts to violate the restrictions specified in “4.4.1.6””

This requirement is addressed by the Windows OS as follows.

As described in the justification text for addressing the Commercial Grade OS Requirement Set “4.4.1.6” requirement, the list of protected roots (consisting of a list of certificate SHA-1 hashes) for the user account resides in the “certificates” value of the following registry key for the user account.

* HKEY\_CURRENT\_USER\Software\Microsoft\SystemCertificates\Root\ProtectedRoots.

Registry keys are named objects that are subject to the auditing due to the Discretionary Access Control policy decisions as shown in Table 1 of the “Addressing 2.1.1.1 “The OS shall enforce a Discretionary Access Control policy on all subjects, all named objects and all access operations among them”” section of paper. The Windows OS Registry Configuration Manager (config.lib) is the responsible resource manager for registry keys. When a specific value of a registry key is changed, the Windows OS Registry Configuration Manager is capable to generate an [Event ID 4657](http://support.microsoft.com/kb/947226) “A registry value was modified” (SE\_AUDITID\_ETW\_REG\_VALUE\_CHANGE\_AUDIT\_value) security audit record for success with the following informational items when available:

* Security ID:
* Account Name:
* Account Domain:
* Logon ID:
* Object Name:
* Object Value Name:
* Handle ID:
* Operation Type:
* Process ID:
* Process Name:
* Old Value Type:
* Old Value:
* New Value Type:
* New Value:.

Consequently, this Commercial Grade OS Requirement Set “4.4.3.2” requirement is met.

# Meeting the “Cryptographic Support Cryptographic Testing Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 3 individual functional requirements under the heading of “Cryptographic Support Cryptographic Testing Functional Requirements”. They are listed as “4.5.1.n”, where n = 1, 2, and 3.

## Addressing 4.5.1.1 “The OS crypto module shall run a suite of self-tests to demonstrate the correct operation of the cryptographic functions in accordance with [FIPS 140-2 Section 4.9] during specific occasions”

The Commercial Grade OS Requirement Set requires the following specific occasions to run the self tests.

1. The initial start-up (on power on);
2. At the request of an authorized administrator (on demand);
3. Under various conditions defined in [FIPS 140-2 Section 4.9];
4. Periodically (at least once a day).

This requirement is addressed by the Windows OS as follows.

We recall from the following section of this paper that the Windows OS provides the FIPS 140-2 validated [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) for applications to use their cryptographic services in the user mode and the kernel mode respectively.

* “Addressing 4.1.1.1 “The OS shall provide a specific list of cryptographic services to applications””.

The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has received FIPS-140-2 (Cert # 892 and Cert # 1008) and the Windows OS kernel security device driver (ksecdd.sys) has received FIPS-140-2 (Cert # 891 and Cert # 1007).

### Windows OS Cryptographic Primitives Library (bcrypt.dll)

As stated in the FIPS-140-2 required [security policy document](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1008.pdf) for the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) on the [NIST Cryptographic Module Validation Program (CMVP) web site](http://csrc.nist.gov/groups/STM/cmvp/index.html), the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) performs the following power-on (start up) self-tests, which are relevant to its FIPS-140-2 validation, when its [DllMain()](http://msdn.microsoft.com/en-us/library/ms682583.aspx) is called by the Windows OS due to a loading attempt from a Windows OS process:

* SHA-1 hash Known Answer Test;
* HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 Known Answer Test;
* Triple-DES (i.e. TDEA) encrypt/decrypt EBC Known Answer Test;
* Triple-DES (i.e. TDEA) encrypt/decrypt CBC Known Answer Test;
* AES-128, AES-192, AES-256 encrypt/decrypt EBC Known Answer Test;
* AES-128, AES-192, AES-256 encrypt/decrypt CBC Known Answer Test;
* AES-128, AES-192, AES-256 encrypt/decrypt CFB with 8-bit feedback Known Answer Test;
* AES-128, AES-192, AES-256 encrypt/decrypt CCM Known Answer Test;
* DSA sign/verify test;
* RSA (i.e. rDSA) sign and verify test;
* DH (i.e. Diffie-Hellman Finite Field-based key agreement algorithm) secret agreement Known Answer Test;
* ECDSA sign/verify test;
* ECDH (i.e. EC Diffie-Hellman Elliptic Curve-based key agreement algorithm) secret agreement Known Answer Test;
* FIPS 186-2 DSA random generator (i.e. FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)]) Known Answer Test;
* NIST SP 800-90 AES-256 based counter mode random generator (i.e. AES CTR\_DRBG) Known Answer Test.

Hence, the occasion a) of this Commercial Grade OS Requirement Set “4.5.1.1” requirement is met for the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) implementations of the aforementioned algorithms.

Note that, currently, a Known Answer Test for the implementation of the NIST SP 800-90 dual elliptic curve deterministic random-number generator (EC\_DRGB) algorithm does not occur in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll).

If any of the above power-on (start up) self-tests fails, the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx)’s [DllMain()](http://msdn.microsoft.com/en-us/library/ms682583.aspx) returns FALSE to the Windows OS. The Windows OS process that attempts to load the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) into itself does not receive a module handle for referencing the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll), and hence no services of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) are available to the process.

In the case where the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) attempts to load the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) through its wrapper ncrypt.dll, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5056](http://support.microsoft.com/kb/947226) “A cryptographic self test was performed” (SE\_AUDITID\_ETW\_NCRYPT\_SELF\_TEST\_EVENT\_value) security audit record for success or failure. A success security audit record is resulted if all the above power-on (start up) self-tests are successful, and both bcrypt.dll and ncrypt.dll are loaded into the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll). A failure security audit record is resulted if any of the above power-on (start up) self-tests fails, and bcrypt.dll and ncrypt.dll are not loaded into the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx).

The FIPS-140-2 required [security policy document](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1008.pdf) also states that the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) performs pair-wise consistency checks upon each invocation of the key-pair generation and the key-pair import for the following algorithms as one of the conditional tests defined in FIPS 140-2 Section 4.9.2.

* DSA;
* RSA (i.e. rDSA);
* ECDSA;
* ECDH (i.e. EC Diffie-Hellman Elliptic Curve-based key agreement algorithm).

We recall that the final step of the key-pair generation or import occurs in [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) and [BCryptImportKeyPair()](http://msdn.microsoft.com/en-us/library/aa375472(VS.85).aspx) respectively.

The FIPS-140-2 required [security policy document](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1008.pdf) also states that the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) performs a continuous RNG test in each of the following implemented RNGs as one of the conditional tests defined in FIPS 140-2 Section 4.9.2.

* FIPS 186-2 DSA random generators (i.e. FIPS 186-2 [(x-Change Notice),(SHA-1)] and FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)]);
* the NIST SP 800-90 dual elliptic curve deterministic random-number generator (EC\_DRGB) algorithm.

Hence, the occasion c) of this Commercial Grade OS Requirement Set “4.5.1.1” requirement is met for the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) implementations of the above mentioned algorithms.

Note that, currently, a continuous RNG test for the implementation of the NIST SP 800-90 AES-256 based counter mode random generator (i.e. AES CTR\_DRBG) does not occur in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll).

In the case where the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) is loaded into the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) through its wrapper ncrypt.dll, the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) generates the following security audit records.

* An [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Key failed pair wise consistency check” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_PAIR\_WISE\_CHECK\_FAILURE) security audit record for failure when
  + the pair-wise consistency check for one of the following algorithm implementations in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) fails:
    - DSA;
    - RSA (i.e. rDSA);
    - ECDSA;
    - ECDH (i.e. EC Diffie-Hellman Elliptic Curve-based key agreement algorithm).
* An [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Random number generation failed FIPS-140 pre-hash check” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_RNG\_FIPS\_140\_FAILURE) security audit record for failure when
  + the continuous RNG test for one of the following algorithm implementations in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) fails:
    - FIPS 186-2 DSA random generators (i.e. FIPS 186-2 [(x-Change Notice),(SHA-1)] and FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)]);
    - the NIST SP 800-90 dual elliptic curve deterministic random-number generator (EC\_DRGB) algorithm.

In any case, when the pair-wise consistency checks fail, the corresponding [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) and [BCryptImportKeyPair()](http://msdn.microsoft.com/en-us/library/aa375472(VS.85).aspx) produce an error status and no valid handle for the private key is returned.

Also, in any case, when the continuous RNG test fails, the corresponding [BCryptGenRandom()](http://msdn.microsoft.com/en-us/library/aa375458(VS.85).aspx) for the specific algorithms, namely [BCRYPT\_RNG\_FIPS186\_DSA\_ALGORITHM](http://msdn.microsoft.com/en-us/library/aa375534(VS.85).aspx) or [BCRYPT\_RNG\_DUAL\_EC\_ALGORITHM](http://msdn.microsoft.com/en-us/library/aa375534(VS.85).aspx), produce an error status and no random number buffer is output.

Note that the [BCRYPT\_RNG\_FIPS186\_DSA\_ALGORITHM](http://msdn.microsoft.com/en-us/library/aa375534(VS.85).aspx) identifier refers to the FIPS 186-2 DSA random generator (i.e. FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)]), and the [BCRYPT\_RNG\_DUAL\_EC\_ALGORITHM](http://msdn.microsoft.com/en-us/library/aa375534(VS.85).aspx) identifier refers to the NIST SP 800-90 dual elliptic curve deterministic random-number generator (EC\_DRGB) algorithm.

After the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has been loaded into a Windows OS process, it would not be able to receive a request of an authorized administrator (on demand), as stated in the occasion b) of this Commercial Grade OS Requirement Set “4.5.1.1” requirement, to re-perform the power-on (start up) self-tests again. Similarly, after the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) has been loaded into a Windows OS process, it does not re-perform the power-on (start up) self-tests periodically (at least once a day), as stated in the occasion d) of this Commercial Grade OS Requirement Set “4.5.1.1” requirement.

However, a local subject can start (on demand) a new Windows OS process that attempts to load the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll). As the loading occurs, the Windows OS calls the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx)’s [DllMain()](http://msdn.microsoft.com/en-us/library/ms682583.aspx). Consequently, the power-on (start up) self-tests are performed.

The same effect can be achieved by an administrator, who stops and restarts the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) (on demand) for causing the restarted [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) to load the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll).

### Windows OS kernel security device driver (ksecdd.sys)

As stated in the FIPS-140-2 required [security policy document](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) for the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) on the [NIST Cryptographic Module Validation Program (CMVP) web site](http://csrc.nist.gov/groups/STM/cmvp/index.html), the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) performs the following power-on (start up) self-tests, which are relevant to its FIPS-140-2 validation, when its [DriverEntry()](http://msdn.microsoft.com/en-us/library/ms795702.aspx) is called by the Windows OS after the Windows OS winload boot application has loaded it into the memory, as explained in the “Integrity check conducted by the Windows OS boot winload application” section of this paper.

* SHA-1 hash Known Answer Test;
* HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 Known Answer Test;
* Triple-DES (i.e. TDEA) encrypt/decrypt EBC Known Answer Test;
* Triple-DES (i.e. TDEA) encrypt/decrypt CBC Known Answer Test;
* AES-128, AES-192, AES-256 encrypt/decrypt EBC Known Answer Test;
* AES-128, AES-192, AES-256 encrypt/decrypt CBC Known Answer Test;
* AES-128, AES-192, AES-256 encrypt/decrypt CFB with 8-bit feedback Known Answer Test;
* AES-128, AES-192, AES-256 GMAC Known Answer Test;
* AES-128, AES-192, AES-256 encrypt/decrypt CCM Known Answer Test;
* RSA (i.e. rDSA) sign and verify test;
* DH (i.e. Diffie-Hellman Finite Field-based key agreement algorithm) secret agreement Known Answer Test;
* ECDSA sign/verify test;
* ECDH (i.e. EC Diffie-Hellman Elliptic Curve-based key agreement algorithm) secret agreement Known Answer Test;
* FIPS 186-2 DSA random generator (i.e. FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)]) Known Answer Test;
* NIST SP 800-90 AES-256 based counter mode random generator (i.e. AES CTR\_DRBG) Known Answer Test.

If any of the above power-on (start up) self-tests fails, the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) stops (i.e. bugchecks) the Windows OS machine and displays the following message on the screen with the indication of the ksecdd.sys [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf).

* “The cryptographic subsystem failed a mandatory algorithm self-test during bootstrap”.

The FIPS-140-2 required [security policy document](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) also states that the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) performs pair-wise consistency checks upon each invocation of the key-pair generation and the key-pair import for the following algorithms as one of the conditional tests defined in FIPS 140-2 Section 4.9.2.

* RSA (i.e. rDSA);
* ECDSA;
* ECDH (i.e. EC Diffie-Hellman Elliptic Curve-based key agreement algorithm).

We recall that the final step of the key-pair generation or import occurs in [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) and [BCryptImportKeyPair()](http://msdn.microsoft.com/en-us/library/aa375472(VS.85).aspx) respectively.

The FIPS-140-2 required [security policy document](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) also states that the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) performs a continuous RNG test in each of the following implemented RNGs as one of the conditional tests defined in FIPS 140-2 Section 4.9.2.

* FIPS 186-2 DSA random generator (i.e. FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)]);
* the NIST SP 800-90 dual elliptic curve deterministic random-number generator (EC\_DRGB) algorithm.

Hence, the occasion c) of this Commercial Grade OS Requirement Set “4.5.1.1” requirement is met for the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) implementations of the above mentioned algorithms.

Note that, currently, a continuous RNG test for the implementation of the NIST SP 800-90 AES-256 based counter mode random generator (i.e. AES CTR\_DRBG) does not occur in the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys).

In any case, when the pair-wise consistency checks fail, the corresponding [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) and [BCryptImportKeyPair()](http://msdn.microsoft.com/en-us/library/aa375472(VS.85).aspx) produce an error status and no valid handle for the private key is returned.

Also, in any case, when the continuous RNG test fails, the corresponding [BCryptGenRandom()](http://msdn.microsoft.com/en-us/library/aa375458(VS.85).aspx) for the specific algorithms, namely [BCRYPT\_RNG\_FIPS186\_DSA\_ALGORITHM](http://msdn.microsoft.com/en-us/library/aa375534(VS.85).aspx) or [BCRYPT\_RNG\_DUAL\_EC\_ALGORITHM](http://msdn.microsoft.com/en-us/library/aa375534(VS.85).aspx), produce an error status and no random number buffer is output.

Note that the [BCRYPT\_RNG\_FIPS186\_DSA\_ALGORITHM](http://msdn.microsoft.com/en-us/library/aa375534(VS.85).aspx) identifier refers to the FIPS 186-2 DSA random generator (i.e. FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)]), and the [BCRYPT\_RNG\_DUAL\_EC\_ALGORITHM](http://msdn.microsoft.com/en-us/library/aa375534(VS.85).aspx) identifier refers to the NIST SP 800-90 dual elliptic curve deterministic random-number generator (EC\_DRGB) algorithm.

After the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) has been loaded into the memory, it exists in the kernel mode and its cryptographic services are available to kernel mode callers. The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) is not capable to receive a request of an authorized administrator (on demand), as stated in the occasion b) of this Commercial Grade OS Requirement Set “4.5.1.1” requirement, to re-perform the power-on (start up) self-tests again. It also does not re-perform the power-on (start up) self-tests periodically (at least once a day), as stated in the occasion d) of this Commercial Grade OS Requirement Set “4.5.1.1” requirement.

Consequently, this Commercial Grade OS Requirement Set “4.5.1.1” requirement is met, except the occasions b) and d) for re-performing the power-on (start up) self-tests again after the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) have passed the power-on (start up) self-tests once and have been successfully loaded.

## Addressing 4.5.1.2 “If any of the cryptographic self-tests fail, the OS shall react as required by [FIPS 140-2 Section 4.9] for failing a self-test”

This requirement is addressed by the Windows OS as follows.

As described in the justification text for addressing the Commercial Grade OS Requirement Set “4.5.1.1” requirement, the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) behave as follows when any of the cryptographic self-tests of the Commercial Grade OS Requirement Set “4.5.1.1” requirement fails.

If any of its power-on (start up) self-tests fails, the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx)’s [DllMain()](http://msdn.microsoft.com/en-us/library/ms682583.aspx) returns FALSE to the Windows OS. The Windows OS process that attempts to load the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) into itself does not receive a module handle for referencing the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll), and hence no services of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) are available to the process.

In the case where the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) attempts to load the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) through its wrapper ncrypt.dll, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5056](http://support.microsoft.com/kb/947226) “A cryptographic self test was performed” (SE\_AUDITID\_ETW\_NCRYPT\_SELF\_TEST\_EVENT\_value) security audit record for success or failure. A failure security audit record is resulted if any of the power-on (start up) self-tests fails, and bcrypt.dll and ncrypt.dll are not loaded into the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx).

If any of its power-on (start up) self-tests fails, the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) stops (i.e. bugchecks) the Windows OS machine and displays the following message on the screen with the indication of the ksecdd.sys ([Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf)).

* “The cryptographic subsystem failed a mandatory algorithm self-test during bootstrap”.

In the case where the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) is loaded into the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) through its wrapper ncrypt.dll, the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) generates the following security audit records.

* An [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Key failed pair wise consistency check” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_PAIR\_WISE\_CHECK\_FAILURE) security audit record for failure when
  + the pair-wise consistency checks for one of the following algorithm implementations in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) fails:
    - DSA;
    - RSA (i.e. rDSA);
    - ECDSA;
    - ECDH (i.e. EC Diffie-Hellman Elliptic Curve-based key agreement algorithm).
* An [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Random number generation failed FIPS-140 pre-hash check” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_RNG\_FIPS\_140\_FAILURE) security audit record for failure when
  + the continuous RNG test for one of the following algorithm implementations in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) fails:
    - FIPS 186-2 DSA random generators (i.e. FIPS 186-2 [(x-Change Notice),(SHA-1)] and FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)]);
    - the NIST SP 800-90 dual elliptic curve deterministic random-number generator (EC\_DRGB) algorithm.

In any case, when the pair-wise consistency checks fail, the corresponding [BCryptFinalizeKeyPair()](http://msdn.microsoft.com/en-us/library/aa375439(VS.85).aspx) and [BCryptImportKeyPair()](http://msdn.microsoft.com/en-us/library/aa375472(VS.85).aspx) of both the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) produce an error status and no valid handle for the private key is returned.

Also, in any case, when the continuous RNG test fails, the corresponding [BCryptGenRandom()](http://msdn.microsoft.com/en-us/library/aa375458(VS.85).aspx) of both the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) for the specific algorithms, namely [BCRYPT\_RNG\_FIPS186\_DSA\_ALGORITHM](http://msdn.microsoft.com/en-us/library/aa375534(VS.85).aspx) or [BCRYPT\_RNG\_DUAL\_EC\_ALGORITHM](http://msdn.microsoft.com/en-us/library/aa375534(VS.85).aspx), produce an error status and no random number buffer is output.

Note that the [BCRYPT\_RNG\_FIPS186\_DSA\_ALGORITHM](http://msdn.microsoft.com/en-us/library/aa375534(VS.85).aspx) identifier refers to the FIPS 186-2 DSA random generator (i.e. FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)]), and the [BCRYPT\_RNG\_DUAL\_EC\_ALGORITHM](http://msdn.microsoft.com/en-us/library/aa375534(VS.85).aspx) identifier refers to the NIST SP 800-90 dual elliptic curve deterministic random-number generator (EC\_DRGB) algorithm.

Consequently, this Commercial Grade OS Requirement Set “4.5.1.2” requirement is addressed.

## Addressing 4.5.1.3 “The OS crypto module shall run a set of specific additional RNG tests”

The Commercial Grade OS Requirement Set requires the following specific additional RNG tests.

1. All known answer RNG tests of [NIST SP 800-90] upon demand and upon power on;
2. The noise sources shall undergo a test at startup to ensure that constant data is not being produced.

This requirement is addressed by the Windows OS as follows.

The [NIST SP 800-90](http://csrc.nist.gov/publications/nistpubs/800-90/SP800-90revised_March2007.pdf) defines three known answer tests in its section “11.3.1” as follows:

* Section “11.3.2”: Testing the Instantiate Function;
* Section “11.3.3”: Testing the Generate Function;
* Section “11.3.4”: Testing the Reseed Function.

Other than those which are relevant to their FIPS-140-2 validations as mentioned in the justification text for addressing the Commercial Grade OS Requirement Set “4.5.1.1” requirement, the power-on (start up) self-tests of both the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) additionally include

* Section “11.3.2”: Testing the Instantiate Function:
  + the known answer test on the instantiate function using the CTR\_DRBG\_Instantiate\_algorithm with the CTR\_DRBG Instantiate Process according to the [NIST SP 800-90](http://csrc.nist.gov/publications/nistpubs/800-90/SP800-90revised_March2007.pdf) Section 10.2.1.3.1 “The Process Steps for Instantiation When Full Entropy is Available for the Entropy Input, and a Derivation Function is Not Used”, where the “AES-128 encrypt EBC” is the Block\_Encrypt in the corresponding CTR\_DRBG Update Process;
* Section “11.3.3”: Testing the Generate Function:
  + the known answer test on the generate function using the CTR\_DRBG\_Generate\_algorithm with the CTR\_DRBG Generate Process according to the [NIST SP 800-90](http://csrc.nist.gov/publications/nistpubs/800-90/SP800-90revised_March2007.pdf) Section 10.2.1.5.1 “The Process Steps for Generating Pseudorandom Bits When a Derivation Function is Not Used for the DRBG Implementation”, where the “AES-128 encrypt EBC” is the Block\_Encrypt in the corresponding CTR\_DRBG Generate Process and CTR\_DRBG Update Process.

Like the other power-on (start up) self tests, if the known answer test on the instantiate function or the known answer test on the generate function fails in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll), then the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) is not loaded by the Windows OS. Similarly, if the known answer test on the instantiate function or the known answer test on the generate function fails in the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys), then the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) is not loaded by the Windows OS.

While the known answer test on the instantiate function and the known answer test on the generate function occur during the power-on (start up) of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys), they do not occur upon demand. The [NIST SP 800-90](http://csrc.nist.gov/publications/nistpubs/800-90/SP800-90revised_March2007.pdf) Section 11.3.2 “Testing the Instantiate Function” and Section 11.3.3 “Testing the Generate Function” only require the corresponding known answer test to run during the power-on (start up).

Both the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) conduct the following known answer test of the reseed function every time after a fresh entropy input is obtained from the noise source but before actually using the entropy input in the CTR\_DRBG\_Reseed\_algorithm with the CTR\_DRBG Reseed Process.

* Section “11.3.4”: Testing the Reseed Function:
  + the known answer test on the reseed function using the CTR\_DRBG\_Reseed\_algorithm with the CTR\_DRBG Reseed Process according to the [NIST SP 800-90](http://csrc.nist.gov/publications/nistpubs/800-90/SP800-90revised_March2007.pdf) Section 10.2.1.4.1 “The Process Steps for Reseeding When Full Entropy is Available for the Entropy Input, and a Derivation Function is Not Used”, where the “AES-128 encrypt EBC” is the Block\_Encrypt in the corresponding CTR\_DRBG Update Process.

In both the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys), the known answer test of the reseed function occurs regardless which one of the following three RNG algorithms is selected.

* [BCRYPT\_RNG\_FIPS186\_DSA\_ALGORITHM (FIPS186DSARNG)](http://msdn.microsoft.com/en-us/library/aa375534.aspx)
  + This is the random-number generator algorithm from FIPS 186-2 DSA (Digital Signature Algorithm);
  + The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) have received the following corresponding RNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for this algorithm:
    - Cert # 435 for
      * FIPS 186-2 [(x-Change Notice),(SHA-1)];
      * FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)];
* [BCRYPT\_RNG\_DUAL\_EC\_ALGORITHM (DUALECRNG)](http://msdn.microsoft.com/en-us/library/aa375534.aspx)
  + This is the NIST SP 800-90 dual elliptic curve deterministic random-number generator (EC\_DRGB) algorithm.
  + The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) have not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for this EC\_DRGB algorithm.
* [BCRYPT\_RNG\_ALGORITHM (RNG)](http://msdn.microsoft.com/en-us/library/aa375534.aspx)
  + This is the NIST SP 800-90 AES counter mode deterministic random-number generator (AES CTR\_DRBG) algorithm.
  + The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) have not received the corresponding DRNG algorithm validation certificates from [NIST CAVP](http://csrc.nist.gov/groups/STM/cavp/index.html) for this AES CTR\_DRBG algorithm.

If the known answer test on the reseed function fails, another fresh entropy input is obtained from the noise source before trying the known answer test on the reseed function again. The previous entropy input from the noise source that has failed the known answer test is ignored. The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) would obtain a fresh entropy input from the noise source and try the known answer test again up to five times, before returning a [STATUS\_INTERNAL\_ERROR](http://msdn.microsoft.com/en-us/library/cc704588.aspx) status code to the subject, who requests a random number.

In the case where the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) is loaded into the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) through its wrapper ncrypt.dll, the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) generates the following security audit record.

* An [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Random number generation failed FIPS-140 pre-hash check” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_RNG\_FIPS\_140\_FAILURE) security audit record for failure when
  + the known answer test on the reseed function in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) fails.

After passing the known answer test on the reseed function, the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) compare the current entropy input from the noise source with the last entropy input from the noise source which has been saved during the last successful known answer test on the reseed function and entropy input comparison test. This entropy input comparison test ensures that the noise source is not producing constant data, as stated in the additional test b) of this requirement. If the entropy input comparison test fails, the current entropy input from the noise source is ignored. The [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) would obtain a fresh entropy input from the noise source and try the known answer test and entropy input comparison test again up to five times, before returning a [STATUS\_INTERNAL\_ERROR](http://msdn.microsoft.com/en-us/library/cc704588.aspx) status code to the subject, who requests a fresh random number.

In the case where the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) is loaded into the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) through its wrapper ncrypt.dll, the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) generates the following security audit record.

* An [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Random number generation failed FIPS-140 pre-hash check” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_RNG\_FIPS\_140\_FAILURE) security audit record for failure when
  + the noise source entropy input comparison in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) fails.

Consequently, the additional test specified in the b) part of this requirement is addressed.

Overall, except the “upon demand” aspect stated in the a) part, this Commercial Grade OS Requirement Set “4.5.1.3” requirement is addressed.

# Meeting the “Cryptographic Support Cryptographic Testing Management Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual management requirement under the heading of “Cryptographic Support Cryptographic Testing Management Requirements”. It is listed as “4.5.2.1”.

## Addressing 4.5.2.1 “The OS shall provide the ability for an authorized administrator to run a suite of crypto module self-tests and RNG tests”

This requirement is addressed by the Windows OS as follows.

Recall that the known answer test on the instantiate function and the known answer test on the generate function are described in the justification text for addressing the Commercial Grade OS Requirement Set “4.5.1.3” requirement. While the known answer test on the instantiate function and the known answer test on the generate function occur during the power-on (start up) of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys), they do not occur upon demand. The [NIST SP 800-90](http://csrc.nist.gov/publications/nistpubs/800-90/SP800-90revised_March2007.pdf) Section 11.3.2 “Testing the Instantiate Function” and Section 11.3.3 “Testing the Generate Function” only require the corresponding known answer test to run during the power-on (start up). After they have been loaded by the Windows OS and have started, they would not be able to receive a request of an authorized administrator (on demand), as stated in the occasion b) of the Commercial Grade OS Requirement Set “4.5.1.1” requirement, to re-perform the power-on (start up) self-tests (including specifically the known answer test on the instantiate function and the known answer test on the generate function) again.

Recall that the known answer test on the reseed function and the entropy input comparison test are described in the justification text for addressing the Commercial Grade OS Requirement Set “4.5.1.3” requirement. These two tests occur whenever the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) have received a subject’s request for a fresh random number. Therefore, supporting a request of an authorized administrator (on demand) to conduct the known answer test on the reseed function and the entropy input comparison test seems unnecessary.

Consequently, except the “upon demand” aspect, this Commercial Grade OS Requirement Set “4.5.1.3” requirement is addressed.

# Meeting the “Cryptographic Support Cryptographic Testing Audit Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual audit requirement under the heading of “Cryptographic Support Cryptographic Testing Audit Requirements”. It is listed as “4.5.3.1”.

## Addressing 4.5.3.1 “The OS shall audit failure of any crypto module self-tests and RNG tests”

This requirement is addressed by the Windows OS as follows.

As described in the justification text for addressing the Commercial Grade OS Requirement Set “4.5.1.1” requirement and the Commercial Grade OS Requirement Set “4.5.1.3” requirement, the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) and the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) behave as follows when any of the cryptographic self-tests of the Commercial Grade OS Requirement Set “4.5.1.1” requirement or any of the RNG tests the Commercial Grade OS Requirement Set “4.5.1.3” requirement fails.

If any of its power-on (start up) self-tests fails, the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx)’s [DllMain()](http://msdn.microsoft.com/en-us/library/ms682583.aspx) returns FALSE to the Windows OS. The Windows OS process that attempts to load the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) into itself does not receive a module handle for referencing the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll), and hence no services of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) are available to the process.

In the case where the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) attempts to load the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) through its wrapper ncrypt.dll, the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) generates an [Event ID 5056](http://support.microsoft.com/kb/947226) “A cryptographic self test was performed” (SE\_AUDITID\_ETW\_NCRYPT\_SELF\_TEST\_EVENT\_value) security audit record for success or failure. A failure security audit record is resulted if any of the power-on (start up) self-tests fails, and bcrypt.dll and ncrypt.dll are not loaded into the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx).

If any of its power-on (start up) self-tests fails, the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) stops (i.e. bugchecks) the Windows OS machine and displays the following message on the screen with the indication of the ksecdd.sys ([Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf)).

* “The cryptographic subsystem failed a mandatory algorithm self-test during bootstrap”.

In the case where the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) is loaded into the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) through its wrapper ncrypt.dll, the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) generates the following security audit records.

* An [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Key failed pair wise consistency check” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_PAIR\_WISE\_CHECK\_FAILURE) security audit record for failure when
  + the pair-wise consistency checks for one of the following algorithm implementations in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) fails:
    - DSA;
    - RSA (i.e. rDSA);
    - ECDSA;
    - ECDH (i.e. EC Diffie-Hellman Elliptic Curve-based key agreement algorithm).
* An [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Random number generation failed FIPS-140 pre-hash check” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_RNG\_FIPS\_140\_FAILURE) security audit record for failure when
  + the continuous RNG test for one of the following algorithm implementations in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) fails:
    - FIPS 186-2 DSA random generators (i.e. FIPS 186-2 [(x-Change Notice),(SHA-1)] and FIPS 186-2 General Purpose [(x-Change Notice),(SHA-1)]);
    - the NIST SP 800-90 dual elliptic curve deterministic random-number generator (EC\_DRGB) algorithm.
* An [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Random number generation failed FIPS-140 pre-hash check” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_RNG\_FIPS\_140\_FAILURE) security audit record for failure when
  + the known answer test on the reseed function in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) fails.
* An [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Random number generation failed FIPS-140 pre-hash check” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_RNG\_FIPS\_140\_FAILURE) security audit record for failure when
  + the noise source entropy input comparison in the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) fails.

Consequently, this Commercial Grade OS Requirement Set “4.5.3.1” requirement is addressed.

# Meeting the “Information Protection Residual Information Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 2 individual functional requirements under the heading of “Information Protection Residual Information Functional Requirements”. They are listed as “5.1.1.n”, where n = 1 and 2.

## Addressing 5.1.1.1 “The OS shall ensure that any previous information content of a resource is made unavailable upon either the allocation of the resource to or the de-allocation of the resource from all non cryptographic objects”

This requirement is addressed by the Windows OS as follows.

The Windows OS ensures that any previous information content of a resource is unavailable upon the allocation of the resource to non cryptographic objects, when they are requested by a subject. As the resource is potentially accessible to a standard user mode requester, the Windows OS ensures the resource does not contain any previous information content.

Objects are initialized from

* disk storage through a supported local file system, or
* memory.

### Local file system files

In the case of disk storage based object initialization, a local file system, supported as parts of the Windows OS, implements the [valid data length (VDL)](http://msdn.microsoft.com/en-us/library/aa364404.aspx) concept. By default, the local file system is [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx). However, it could also be [fastfat.sys](http://technet.microsoft.com/en-us/library/cc776720.aspx) or [exfat.sys](http://msdn.microsoft.com/en-us/library/aa914353.aspx).

* The valid data length (VDL) is the length (in bytes) of a data stream in a file that is actually written. This value is always less than or equal to the file data stream size.

In its handler of the [IRP\_MJ\_WRITE](http://msdn.microsoft.com/en-us/library/ms806163.aspx) IO function, a local file system (namely, [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx), [fastfat.sys](http://technet.microsoft.com/en-us/library/cc776720.aspx), or [exfat.sys](http://msdn.microsoft.com/en-us/library/aa914353.aspx)) ensures that the region between the new requested write position (byte offset) and the new valid data length is zeroed when it has to extend the valid data length for fulfilling a specific write request of its caller. This action ensures that the range between the start and the current valid data length of a file data stream is filled with either zeros or the actual supplied data of the write requester.

In the case of NTFS, an authorized subject, possessing the [SeManageVolumePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK), may request the ntfs.sys file system driver to extend the valid data length of a specified file data stream where [there is no explicit zeroing between the current write position (byte offset) and the new valid data length](http://msdn.microsoft.com/en-us/library/aa365544(VS.85).aspx) for the performance reason. By default, the [SeManageVolumePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) is granted only to the “Administrators” group.

In its handler of the [IRP\_MJ\_READ](http://msdn.microsoft.com/en-us/library/ms806158.aspx) IO function, a local file system (namely, [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx), [fastfat.sys](http://technet.microsoft.com/en-us/library/cc776720.aspx), or [exfat.sys](http://msdn.microsoft.com/en-us/library/aa914353.aspx)) ensures that

* if the current requested read position is beyond the end of file, then the local file system returns no data but the [STATUS\_END\_OF\_FILE](http://msdn.microsoft.com/en-us/library/cc704588.aspx) error code;
* if the current requested read position is within the end of file, but the requested read range reaches beyond the end of file, then the local file system truncates the read range to the end of file for the read requester subject;
* if the current requested read position is beyond the current valid data length, then the local file system returns zeros in the read requester subject’s buffer;
* if the requested read range reaches beyond the current valid data length but within the end of file, and the current requested read position is within the current valid data length, then the local file system truncates the read range to the current valid data length for the read requester subject.

These actions ensure that the read requester subject’s buffer is filled with either the actual data of the requested file data stream or zeros, until the end of file is reached.

### Windows OS resource managers managing objects

In the case of memory based object initialization, every Windows OS resource manager is responsible for the memory based object initialization of its objects. The resource manager allocates a piece of memory of a specific size for holding an object resource. Through the interaction with the resource manager’s interfaces, either the data being requested by an authorized caller subject or the data being supplied by an authorized caller subject is intended for a temporary residence in an object resource. As soon as the piece of memory is allocated, the resource manager fills it with zeros. As a result, after copying the relevant data for an authorized caller subject into the allocated memory, the corresponding object resource consists of either zeros or the relevant data for an authorized caller subject.

### Windows OS microkernel managing frames

The Windows OS microkernel component initializes and maintains the following Windows OS thread-specific frames for each thread being executed by the hardware.

* Trap frame:
  + It is used to handle a trap for saving and restoring the execution state of the thread that has been interrupted, including program counters, volatile and nonvolatile registers, and other information, when a user mode kernel mode context switching occurs.
* Context frame:
  + It is used for saving and restoring the execution state of the previous thread and new thread respectively when a thread switching occurs.
* Exception frame:
  + It is used for recording nonvolatile registers when handling an exception for the specific thread.

During their initialization, these frames are necessarily zeroed before they are used. Because of the frame saving and restoring, information contained in registers and counters for the previous thread is inaccessible to the new thread after the occurrence of a user mode kernel mode context switching or thread switching.

### Windows OS window terminal sessions

The Windows OS local session manager (lsm.exe) initializes and manages Windows OS window terminal sessions. An established session can be disconnected and reconnected later on upon user re-authentication. The Windows OS local session manager (lsm.exe) and Windows OS remote window terminal service (termsrv.dll) together mediate the authorized access to a disconnected session and the authorized shadowing of a connected session.

On a Windows OS machine, there is only one local window terminal session that is connected to the local monitor and input devices such as keyboard, mouse, and/or others, so that a local interactive user interacts with the session. When the Windows OS local session manager disconnects the current local window terminal session from the local monitor and input devices, the window terminal session becomes disconnected. A disconnected session is not visible and it is not capable to receive input. After the previous local session is disconnected, another session is connected to the local monitor and input devices to become the current local session.

On a Windows OS machine, multiple remote window terminal sessions are possible. The remote window terminal service controls the access to a remote window terminal session. A remote window terminal session is connected to a network stack, which delivers graphical elements of the session to and receives input elements from the authenticated remote interactive user.

The Windows OS remote window terminal service (termsrv.dll), together with the assistance of the Windows OS local session manager (lsm.exe), mediates the authorized access to a disconnected window terminal session and the authorized shadowing of a connected window terminal session. Therefore, the graphical user interfaces of a window terminal session are accessible to authorized interactive users only.

More details on Windows OS window terminal sessions can be found in the following sections of this paper.

* “Human interface device input and video output separation maintained by the Windows OS window manager(s)”;
* “Remote or local window terminal session disconnection”;
* “Remote or local window terminal session reconnection from a (distinct) source window terminal session”.

As a result, the intention of this Commercial Grade OS Requirement Set “5.1.1.1” requirement is addressed.

## Addressing 5.1.1.2 “Objects associated with cryptographic keys and critical cryptographic security parameters shall be destroyed as described in the Commercial Grade OS Requirement Set “4.4.1.8” requirement”

This requirement is addressed by the Windows OS as follows.

The Commercial Grade OS Requirement Set “4.4.1.8” requirement states that the OS crypto module shall destroy cryptographic keys in accordance with a cryptographic key zeroization method that meets the following requirements.

1. Key zeroization requirements in [FIPS PUB 140-2], “Security Requirements for Cryptographic Modules.”
2. Zeroization of all plaintext cryptographic keys and all other critical cryptographic security parameters shall be immediate and complete.
3. Each intermediate storage area for plaintext cryptographic key /critical cryptographic security parameter (i.e. any storage, such as memory buffers, that is included in the path of such data) shall be zeroized.
4. Zeroization shall be executed by one of the following two selections:
   1. For non-volatile memories other than EEPROM and Flash, the zeroization shall be executed by overwriting three or more times using a different alternating data pattern each time;
   2. For volatile memories and non-volatile EEPROM and Flash memories, the zeroization shall be executed by a single direct overwrite consisting of a pseudo random pattern, followed by a read-verify.

Long lived cryptographic keys or critical cryptographic security parameters are not stored in plaintext, but in an appropriate encrypted form, in a persistent area of the Windows OS. They appear in plaintext while temporarily being copied into a memory buffer only, as they need to be used by a Windows OS cryptographic module. When they are not used, they are encrypted even when they reside in memory rather than in disk storage, using typically, either [CryptProtectMemory()](http://msdn.microsoft.com/en-us/library/aa380262.aspx) or the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) [CryptProtectData()](http://msdn.microsoft.com/en-us/library/aa380261(VS.85).aspx). This action limits the risks that their plaintexts may be leaked when they have to reside in an intermediate storage area that is included in their execution paths. In the case of [CryptProtectMemory()](http://msdn.microsoft.com/en-us/library/aa380262.aspx) , the corresponding encryption/decryption keys are valid only when the Windows OS is running. They are not persisted across Windows OS boots. They exist only in the [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys) without involving any intermediate storage areas. They are freshly generated anew, when ksecdd.sys starts during the Windows OS boot time. They are zeroed, when ksecdd.sys is unloaded due to the Windows OS shutting down. In the case of the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) [CryptProtectData()](http://msdn.microsoft.com/en-us/library/aa380261(VS.85).aspx), the security of the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) protection is described in the following two sections of this paper.

* “Kerberos security provider notifies the local Windows OS data protection API manger”;
* “NTLM security provider notifies the local Windows OS data protection API manger”.

Due to the encryption of long lived cryptographic keys or critical cryptographic security parameters using either [CryptProtectMemory()](http://msdn.microsoft.com/en-us/library/aa380262.aspx) or the [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) [CryptProtectData()](http://msdn.microsoft.com/en-us/library/aa380261(VS.85).aspx) while in memory, the zeroization of the intermediate storage areas in their execution paths as required in c) above is not as critical.

Immediately after they are used, the plaintext cryptographic keys or critical cryptographic security parameters are destroyed by [RtlSecureZeroMemory()](http://msdn.microsoft.com/en-us/library/aa366877.aspx) as required in b) above. [RtlSecureZeroMemory()](http://msdn.microsoft.com/en-us/library/aa366877.aspx) is a [force inline](http://msdn.microsoft.com/en-us/library/z8y1yy88(VS.80).aspx) function that overwrites a memory buffer with zeros once. [RtlSecureZeroMemory()](http://msdn.microsoft.com/en-us/library/aa366877.aspx) is not optimized by the compiler. In the case where the Windows OS runs on an x64 machine, the memory buffer overwrite is achieved by calling the [stosb()](http://msdn.microsoft.com/en-us/library/ywwcatsa.aspx) x64 specific instruction to store only zeros in the buffer. In the case where the Windows OS runs on a non-x64 machine, the memory buffer overwrite is achieved by assigning a zero to every slot in the memory buffer. After the memory buffer overwrite, [RtlSecureZeroMemory()](http://msdn.microsoft.com/en-us/library/aa366877.aspx) returns a pointer to its caller, where the pointer points to the start of the memory buffer. However, it is not a usual practice that a [RtlSecureZeroMemory()](http://msdn.microsoft.com/en-us/library/aa366877.aspx) caller would read the overwritten memory buffer to verify that the buffer contains only zeros, as required in the read-verify action. Due to the [force inline](http://msdn.microsoft.com/en-us/library/z8y1yy88(VS.80).aspx) nature of [RtlSecureZeroMemory()](http://msdn.microsoft.com/en-us/library/aa366877.aspx), the [stosb()](http://msdn.microsoft.com/en-us/library/ywwcatsa.aspx) x64 instruction and the zero assignment always take place. Therefore, the read-verify action as required in d)2) above is not necessary.

As explained in the “Protections provided by the Windows OS key isolation service” section of this paper, the FIPS-140-2 validated [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) takes additional care when it is requested to destroy specific private keys and secret agreement values from the memory.

When [BCryptDestroyKey()](http://msdn.microsoft.com/en-us/library/aa375404(vs.85).aspx) of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) is used to destroy a clear text private key currently existing in the memory, [BCryptDestroyKey()](http://msdn.microsoft.com/en-us/library/aa375404(vs.85).aspx) does not only overwrite the memory buffer holding the private key with zeros as explained in the above, [BCryptDestroyKey()](http://msdn.microsoft.com/en-us/library/aa375404(vs.85).aspx) also verifies that every slot of the overwritten memory buffer is indeed zero. If [BCryptDestroyKey()](http://msdn.microsoft.com/en-us/library/aa375404(vs.85).aspx) is used within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx), a failure of this read-zero-verify action further results in the generation of an [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Failed to zero secret data” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_ZERO\_DATA\_FAILURE) security audit record for failure.

Similarly, when [BCryptDestroySecret()](http://msdn.microsoft.com/en-us/library/aa375407(VS.85).aspx)of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) is used to destroy a calculated clear text secret agreement value currently existing in the memory, [BCryptDestroySecret()](http://msdn.microsoft.com/en-us/library/aa375407(VS.85).aspx) does not only overwrite the memory buffer holding the calculated secret agreement value with zeros as explained in the above, [BCryptDestroySecret()](http://msdn.microsoft.com/en-us/library/aa375407(VS.85).aspx) also verifies that every slot of the overwritten memory buffer is indeed zero. If [BCryptDestroySecret()](http://msdn.microsoft.com/en-us/library/aa375407(VS.85).aspx) is used within the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx), a failure of this read-zero-verify action further results in the generation of an [Event ID 5057](http://support.microsoft.com/kb/947226) “A cryptographic primitive operation failed (in the case of) Failed to zero secret data” (SE\_AUDITID\_ETW\_NCRYPT\_PRIMITIVE\_FAILURE.Id/SE\_CNG\_ADT\_ZERO\_DATA\_FAILURE) security audit record for failure.

The zeroization method based on [RtlSecureZeroMemory()](http://msdn.microsoft.com/en-us/library/aa366877.aspx) meets the key zeroization requirements in [FIPS PUB 140-2] as required in a) above. This same method or an equivalent is used by all Windows OS cryptographic modules that have received the FIPS-140-2 validations. These Windows OS cryptographic modules are:

* Windows Vista bcrypt.dll has received FIPS-140-2 (Cert # 892). Windows Server 2008 bcrypt.dll has received FIPS-140-2 (Cert # 1008).
* Windows Vista ksecdd.sys has received FIPS-140-2 (Cert # 891). Windows Server 2008 ksecdd.sys has received FIPS-140-2 (Cert # 1007).
* Windows Vista BitLocker™ components have received FIPS-140-2 (Cert # 947). Windows Server 2008 BitLocker™ components have received FIPS-140-2 (Cert # 1054).
* Windows Vista rsaenh.dll has received FIPS-140-2 (Cert # 893). Windows Server 2008 rsaenh.dll has received FIPS-140-2 (Cert # 1010).
* Windows Vista dssenh.dll has received FIPS-140-2 (Cert # 894). Windows Server 2008 dssenh.dll has received FIPS-140-2 (Cert # 1009).

As a result, the intention of this Commercial Grade OS Requirement Set “5.1.1.2” requirement is addressed.

# Meeting the “Information Protection Residual Information Management Requirements”

In the Commercial Grade OS Requirement Set, there is no individual management requirement under the heading of “Information Protection Residual Information Management Requirements”.

# Meeting the “Information Protection Residual Information Audit Requirements”

In the Commercial Grade OS Requirement Set, there is no individual audit requirement under the heading of “Information Protection Residual Information Audit Requirements”.

# Meeting the “Information Protection Resource Control Functional Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual functional requirement under the heading of “Information Protection Resource Control Functional Requirements”. It is listed as “5.2.1.1”.

## Addressing 5.2.1.1 “The OS shall enforce maximum quotas on the portion of shared persistent storage that individual authorized users can use”

This requirement is addressed by the Windows OS as follows.

In the Windows OS, the [disk (i.e. shared persistent storage) quota facility](http://msdn.microsoft.com/en-us/library/aa363995(VS.85).aspx) is available in every NTFS non read-only volume ([ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx)). The disk space that each file object occupies is charged directly to the user account, who is the owner of the file object. The owner is identified by the “[owner SID](http://msdn.microsoft.com/en-us/library/aa379561(VS.85).aspx)” field in the security descriptor associated with the file object. The total disk space charged to a user account is the sum of the length of all chargeable data streams. Hence, property set streams and resident user data streams affect the user account 's disk quota. Disk quota is not charged for re-parse points, security descriptors, or other metadata that is associated with the file objects. Compressing or decompressing files does not affect the disk space reported for the files. Therefore, disk quota settings on one volume can be compared to settings on another volume. The following list identifies the types of disk quota limits.

* Warning threshold:
  + When configured by an administrator, the [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx) generates an [Event ID 36](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=36&EvtSrc=ntfs&LCID=1033) “A user hit their quota threshold” (IO\_FILE\_QUOTA\_THRESHOLD) soft audit record for the soft audit storage after the current disk space charged to a standard user has exceeded his/her warning [disk quota threshold](http://msdn.microsoft.com/en-us/library/aa365053(VS.85).aspx). The current file system operation is not interrupted.
  + The soft audit storage is the audit storage that does not cause the local machine to prevent subsequent auditable events, except those taken by an administrator, to occur when the audit storage is full (i.e. reaching its maximum size).
  + The soft audit record identifies the user account whose warning [disk quota threshold](http://msdn.microsoft.com/en-us/library/aa365053(VS.85).aspx) has been exceeded and the name of the volume that is enforcing the warning [disk quota threshold](http://msdn.microsoft.com/en-us/library/aa365053(VS.85).aspx).
* Hard quota:
  + When configured by an administrator, the [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx) generates an [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) “A user hit their quota limit on volume X” (IO\_FILE\_QUOTA\_LIMIT) soft audit record for the soft audit storage after the current disk space charged to a standard user has exceeded his/her hard [disk quota limit](http://msdn.microsoft.com/en-us/library/aa365052(VS.85).aspx). The [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx) also raises the [STATUS\_DISK\_FULL](http://msdn.microsoft.com/en-us/library/cc704588.aspx) error code. Therefore, the current file system operation is interrupted.
  + The soft audit storage is the audit storage that does not cause the local machine to prevent subsequent auditable events, except those taken by an administrator, to occur when the audit storage is full (i.e. reaching its maximum size).
  + The soft audit record identifies the user account whose hard [disk quota limit](http://msdn.microsoft.com/en-us/library/aa365052(VS.85).aspx) has been exceeded and the name of the volume that is enforcing the hard [disk quota limit](http://msdn.microsoft.com/en-us/library/aa365052(VS.85).aspx).

The [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx) automatically creates a user quota entry when a user subject first writes to a non read-only volume. Entries that are created automatically are assigned the [default warning disk quota threshold](http://msdn.microsoft.com/en-us/library/aa365027(VS.85).aspx) and the [default hard disk quota limit](http://msdn.microsoft.com/en-us/library/aa365026(VS.85).aspx) values for the volume.

For each non read-only volume, its user quota information entries are stored in the system file named “$Quota” under the “$Extend” NTFS master file table metadata file directory of the volume. Each user quota information entry contains the following elements:

* QuotaSid:
  + It indicates the security identifier (SID) of the specific standard user account or the default user account that this quota information entry applies to;
* QuotaFlags:
  + It contains the control flags associated with this quota information entry;
* QuotaUsed:
  + It indicates the amount of disk space in bytes in this volume that is currently being used by subjects of the user account, identified by the QuotaSid;
* QuotaThreshold:
  + It indicates the disk quota warning threshold in bytes for the user account of the QuotaSid;
* QuotaLimit:
  + It indicates the disk quota hard quota in bytes for the user account of the QuotaSid;
* QuotaChangeTime:
  + It records the time when this quota information entry was last changed;
* QuotaExceededTime:
  + It records the time when user account of the QuotaSid has exceeded either the QuotaThreshold or the QuotaLimit value.

The QuotaUsed value of the user quota information entry for a standard user account is updated under the following conditions.

* The QuotaUsed value is set to zero in its anticipation for a quota charge recalculation by enumerating all the user account’s files currently residing in the volume, when the disk quota facility for the volume needs to repair itself.
* During the recalculation, the quota charges for the user account’s files currently residing in the volume are added up. After enumerating all the user account’s files, the sum becomes the current QuotaUsed value.
* When a file belonging to the user account is deleted, the quota charge for the file is reduced from the current QuotaUsed value.
* When a file belonging to the user account is assigned to another user account as the new owner, the quota charge for the file is reduced from the current QuotaUsed value.
* When a file not belonging to the user account is assigned to the user account as the new owner, the quota charge for the file is added to the current QuotaUsed value. If the current QuotaUsed value exceeds the QuotaLimit, then the new ownership assignment fails with the [STATUS\_DISK\_FULL](http://msdn.microsoft.com/en-us/library/cc704588.aspx) error code and an [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) “A user hit their quota limit” (IO\_FILE\_QUOTA\_LIMIT) soft audit record is generated.
* When a new file with the user account as the new owner is created with a “$DATA” file attribute, the number of bytes in a file record segment ([BytesPerFileRecordSegment](http://msdn.microsoft.com/en-us/library/aa365256(VS.85).aspx)) is added to the current QuotaUsed value. If the current QuotaUsed value exceeds the QuotaLimit, then the new file creation fails with the [STATUS\_DISK\_FULL](http://msdn.microsoft.com/en-us/library/cc704588.aspx) error code and an [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) “A user hit their quota limit” (IO\_FILE\_QUOTA\_LIMIT) soft audit record is generated.
* When a new “$DATA” file attribute of a specific size is created for a file belonging to the user account, the size (after an appropriate rounding up based on the cluster size) is added to the current QuotaUsed value. If the current QuotaUsed value exceeds the QuotaLimit, then the new file attribute creation fails with the [STATUS\_DISK\_FULL](http://msdn.microsoft.com/en-us/library/cc704588.aspx) error code and an [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) “A user hit their quota limit” (IO\_FILE\_QUOTA\_LIMIT) soft audit record is generated.
* When an allocation of a specific size is made to extend an existing “$DATA” file attribute for a file belonging to the user account, the size (after an appropriate rounding up based on the cluster size) is added to the current QuotaUsed value. If the current QuotaUsed value exceeds the QuotaLimit, then the file attribute extension fails with the [STATUS\_DISK\_FULL](http://msdn.microsoft.com/en-us/library/cc704588.aspx) error code and an [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) “A user hit their quota limit” (IO\_FILE\_QUOTA\_LIMIT) soft audit record is generated.
* When an allocation of a specific size is deleted from an existing “$DATA” file attribute for a file belonging to the user account, the size (after an appropriate rounding up based on the cluster size) is reduced to the current QuotaUsed value.
* When a new “$DATA” file attribute, which does not have any more allocation, is deleted from a file belonging to the user account, the number of bytes in a file record segment ([BytesPerFileRecordSegment](http://msdn.microsoft.com/en-us/library/aa365256(VS.85).aspx)) is reduced to the current QuotaUsed value.

To limit the risk of event log (i.e. soft audit storage) flooding, the [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) “A user hit their quota limit” (IO\_FILE\_QUOTA\_LIMIT) is not generated every time when the current QuotaUsed value exceeds the QuotaLimit. When the current QuotaUsed value exceeds the QuotaLimit for the first time, the [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) is generated and the current time is set to the QuotaExceededTime. During the next time when the QuotaUsed value exceeds the QuotaLimit, if the current time lies within the NtfsQuotaNotifyRate value range after the QuotaExceededTime, then the [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) is not generated again. However, the QuotaExceededTime is updated with the current time. In other words, there is at most one [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) occurrence within the NtfsQuotaNotifyRate value range.

Similarly, the [Event ID 36](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=36&EvtSrc=ntfs&LCID=1033) “A user hit their quota threshold” (IO\_FILE\_QUOTA\_THRESHOLD) is not generated every time when the current QuotaUsed value exceeds the QuotaThreshold. When the current QuotaUsed value exceeds the QuotaThreshold for the first time, the [Event ID 36](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=36&EvtSrc=ntfs&LCID=1033) is generated and the current time is set to the QuotaExceededTime. During the next time when the QuotaUsed value exceeds the QuotaThreshold, if the current time lies within the NtfsQuotaNotifyRate value range after the QuotaExceededTime, then the [Event ID 36](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=36&EvtSrc=ntfs&LCID=1033) is not generated again. However, the QuotaExceededTime is updated with the current time. In other words, there is at most one [Event ID 36](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=36&EvtSrc=ntfs&LCID=1033) occurrence within the NtfsQuotaNotifyRate value range.

By default, the NtfsQuotaNotifyRate value is 1 hour (i.e. 3600 sec). An administrator can configure this value via the following registry key value.

* HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem\NtfsQuotaNotifyRate

As a result, this Commercial Grade OS Requirement Set “5.2.1.1” requirement is met.

# Meeting the “Information Protection Resource Control Management Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual management requirement under the heading of “Information Protection Resource Control Management Requirements”. It is listed as “5.2.2.1”.

## Addressing 5.2.2.1 “The OS shall allow only authorized administrators the ability to set maximum quotas on shared persistent storage”

This requirement is addressed by the Windows OS as follows.

The Windows OS IO manager provides the [NtSetQuotaInformationFile()](http://msdn.microsoft.com/en-us/library/ms800959.aspx) interface for an authorized subject to set the user quota information entry for a specific user account in a [FILE\_QUOTA\_INFORMATION](http://msdn.microsoft.com/en-us/library/ms791507.aspx) structure. The Windows OS IO manager requires the [NtSetQuotaInformationFile()](http://msdn.microsoft.com/en-us/library/ms800959.aspx) caller subject to possess the FILE\_WRITE\_DATA access to the targeted NTFS non read-only volume. The Windows OS IO manager handles the [NtSetQuotaInformationFile()](http://msdn.microsoft.com/en-us/library/ms800959.aspx) request by forwarding the user quota information entry in an [IRP\_MJ\_SET\_QUOTA](http://msdn.microsoft.com/en-us/library/ms795856.aspx) to the [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx) which manages the targeted volume. The [IRP\_MJ\_SET\_QUOTA](http://msdn.microsoft.com/en-us/library/ms795856.aspx) handler of the [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx) further requires the [NtSetQuotaInformationFile()](http://msdn.microsoft.com/en-us/library/ms800959.aspx) caller subject to possess the [SeManageVolumePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK). By default, the [SeManageVolumePrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) is granted only to the “Administrators” group. The [IRP\_MJ\_SET\_QUOTA](http://msdn.microsoft.com/en-us/library/ms795856.aspx) handler saves the received user quota information entry in the system file named “$Quota” under the “$Extend” NTFS master file table metadata file directory of the volume.

The Windows OS IO manager also provides the [NtSetVolumeInformationFile()](http://msdn.microsoft.com/en-us/library/ms800930.aspx) interface with the FileFsControlInformation FsInformationClass for an authorized subject to set the default quota information entry in a [FILE\_FS\_CONTROL\_INFORMATION](http://msdn.microsoft.com/en-us/library/ms791509.aspx) structure. The Windows OS IO manager requires the [NtSetVolumeInformationFile()](http://msdn.microsoft.com/en-us/library/ms800930.aspx) caller subject to possess the FILE\_WRITE\_DATA access to the targeted NTFS non read-only volume. The Windows OS IO manager handles the [NtSetVolumeInformationFile()](http://msdn.microsoft.com/en-us/library/ms800930.aspx) request by forwarding the default quota information entry in an [IRP\_MJ\_SET\_VOLUME\_INFORMATION](http://msdn.microsoft.com/en-us/library/ms795843.aspx) to the [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx) which manages the targeted volume. The [IRP\_MJ\_SET\_VOLUME\_INFORMATION](http://msdn.microsoft.com/en-us/library/ms795843.aspx) handler saves the received default quota information entry in the system file named “$Quota” under the “$Extend” NTFS master file table metadata file directory of the volume. By default, the FILE\_WRITE\_DATA access to the targeted NTFS non read-only volume is granted only to the “Administrators” group and the local system.

As a result, this Commercial Grade OS Requirement Set “5.2.2.1” requirement is met.

# Meeting the “Information Protection Resource Control Audit Requirements”

In the Commercial Grade OS Requirement Set, there are 2 individual audit requirements under the heading of “Information Protection Resource Control Audit Requirements”. They are listed as “5.2.3.n”, where n = 1 and 2.

## Addressing 5.2.3.1 “The OS shall provide the ability to audit any request by a user that attempts to exceed the authorized administrator defined shared persistent storage quota”

This requirement is addressed by the Windows OS as follows.

Recalling from the justification text for addressing the Commercial Grade OS Requirement Set “5.2.1.1” requirement, the [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx) generates [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) and [Event ID 36](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=36&EvtSrc=ntfs&LCID=1033) soft audit records as follows.

When configured by an administrator, the [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx) generates an [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) “A user hit their quota limit” (IO\_FILE\_QUOTA\_LIMIT) soft audit record for the soft audit storage after the current disk space charged to a standard user has exceeded his/her hard [disk quota limit](http://msdn.microsoft.com/en-us/library/aa365052(VS.85).aspx). The [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx) also raises the [STATUS\_DISK\_FULL](http://msdn.microsoft.com/en-us/library/cc704588.aspx) error code. Therefore, the current file system operation is interrupted.

The soft audit storage is the audit storage that does not cause the local machine to prevent subsequent auditable events, except those taken by an administrator, to occur when the audit storage is full (i.e. reaching its maximum size).

The soft audit record identifies the user whose hard [disk quota limit](http://msdn.microsoft.com/en-us/library/aa365052(VS.85).aspx) has been exceeded and the name of the volume that is enforcing the hard [disk quota limit](http://msdn.microsoft.com/en-us/library/aa365052(VS.85).aspx).

When configured by an administrator, the [ntfs.sys](http://technet.microsoft.com/en-us/library/cc781134.aspx) generates an [Event ID 36](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=36&EvtSrc=ntfs&LCID=1033) “A user hit their quota threshold” (IO\_FILE\_QUOTA\_THRESHOLD) soft audit record for the soft audit storage after the current disk space charged to a standard user has exceeded his/her warning [disk quota threshold](http://msdn.microsoft.com/en-us/library/aa365053(VS.85).aspx). The current file system operation is not interrupted.

The soft audit record identifies the user whose warning [disk quota threshold](http://msdn.microsoft.com/en-us/library/aa365053(VS.85).aspx) has been exceeded and the name of the volume that is enforcing the warning [disk quota threshold](http://msdn.microsoft.com/en-us/library/aa365053(VS.85).aspx).

To limit the risk of event log (i.e. soft audit storage) flooding, the [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) “A user hit their quota limit” (IO\_FILE\_QUOTA\_LIMIT) is not generated every time when the current QuotaUsed value exceeds the QuotaLimit. When the current QuotaUsed value exceeds the QuotaLimit for the first time, the [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) is generated and the current time is set to the QuotaExceededTime. During the next time when the QuotaUsed value exceeds the QuotaLimit, if the current time lies within the NtfsQuotaNotifyRate value range after the QuotaExceededTime, then the [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) is not generated again. However, the QuotaExceededTime is updated with the current time. In other words, there is at most one [Event ID 37](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=37&EvtSrc=ntfs) occurrence within the NtfsQuotaNotifyRate value range.

Similarly, the [Event ID 36](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=36&EvtSrc=ntfs&LCID=1033) “A user hit their quota threshold” (IO\_FILE\_QUOTA\_THRESHOLD) is not generated every time when the current QuotaUsed value exceeds the QuotaThreshold. When the current QuotaUsed value exceeds the QuotaThreshold for the first time, the [Event ID 36](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=36&EvtSrc=ntfs&LCID=1033) is generated and the current time is set to the QuotaExceededTime. During the next time when the QuotaUsed value exceeds the QuotaThreshold, if the current time lies within the NtfsQuotaNotifyRate value range after the QuotaExceededTime, then the [Event ID 36](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=36&EvtSrc=ntfs&LCID=1033) is not generated again. However, the QuotaExceededTime is updated with the current time. In other words, there is at most one [Event ID 36](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=36&EvtSrc=ntfs&LCID=1033) occurrence within the NtfsQuotaNotifyRate value range.

By default, the NtfsQuotaNotifyRate value is 1 hour (i.e. 3600 sec). An administrator can configure this value via the following registry key value.

* HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem\NtfsQuotaNotifyRate

As a result, this Commercial Grade OS Requirement Set “5.2.3.1” requirement is met.

## Addressing 5.2.3.2 “The OS shall provide the ability to audit the setting of maximum quotas on shared persistent storage”

While the Windows OS IO manager provides the [NtSetQuotaInformationFile()](http://msdn.microsoft.com/en-us/library/ms800959.aspx) interface for an authorized subject to set the user quota information entry for a specific user account and the [NtSetVolumeInformationFile()](http://msdn.microsoft.com/en-us/library/ms800930.aspx) interface for an authorized subject to set the default quota information entry, there is no generation of an audit record, within either interface, that identifies the authorized subject who sets the specific user quota information entry or the default quota information entry.

However, it may not be deemed as a serious security shortcoming because the authorized subject, who can exercise the [NtSetQuotaInformationFile()](http://msdn.microsoft.com/en-us/library/ms800959.aspx) and [NtSetVolumeInformationFile()](http://msdn.microsoft.com/en-us/library/ms800930.aspx) interfaces, has to be a member of the “Administrators” group. In addition, an authorized subject, who possesses the FILE\_READ\_DATA access to the targeted NTFS non read-only volume, can query a specific user quota information entry or the default quota information entry currently being enforced within the targeted NTFS non read-only volume using the [NtQueryQuotaInformationFile()](http://msdn.microsoft.com/en-us/library/ms800891.aspx) or [NtQueryVolumeInformationFile()](http://msdn.microsoft.com/en-us/library/ms800886.aspx) respectively.

Therefore, instead of auditing the setting of the user quota information entry for a specific user account and the setting of the default quota information entry, we recommend (as a workaround of this Commercial Grade OS Requirement Set “5.2.3.2” requirement) that an organization monitors the setting of the user quota information entry for a specific user account and the setting of the default quota information entry of specific targeted NTFS non read-only volumes for changes.

# Meeting the “Information Protection Trusted Initialization~~, Self Testing and Recovery~~ Functional Requirements”

In the Commercial Grade OS Requirement Set, there ~~are 3 individual functional requirements~~ is 1 individual functional requirement under the heading of “Information Protection Trusted Initialization~~, Self Testing and Recovery~~ Functional Requirements”. ~~They are listed as “5.3.1.n”, where n = 1, 2 and 3.~~ It is listed as “5.3.1.3”.

## ~~Addressing 5.3.1.1 “The OS shall run a suite of self tests to demonstrate the correct operation of the OS security functions during specific occasions”~~

~~The Commercial Grade OS Requirement Set requires the following specific occasions to run the self tests.~~

1. ~~The initial start-up;~~
2. ~~At the request of an authorized administrator;~~
3. ~~Periodically (at an authorized administrator defined frequency) during normal operations.~~

~~Unfortunately, the Windows OS itself does not incorporate self tests for its security functions specified in this Commercial Grade OS Requirement Set as parts of its operational features. However, as parts of its servicability and sustained engineering processes, Microsoft continues to conduct unit tests and scenario based tests for the Windows OS after the Windows OS is shipped to customers. Issues arisen during the servicability and sustained engineering processes are investigated and addressed in the next scheduled service pack of the Windows OS.~~

## ~~Addressing 5.3.1.2 “After a failure of a self test, the OS shall ensure a maintenance mode where the ability to return the OS to a secure state is provided”~~

~~This requirement is moot because the Windows OS itself does not incorporate self tests for its security functions specified in this Commercial Grade OS Requirement Set as parts of its operational features.~~

## Addressing 5.3.1.3 “The OS shall verify during initial startup the integrity of executable code that implements access control and cryptographic functionality through the use of the OS system provided cryptographic services”

This requirement is addressed by the Windows OS as follows.

The Windows OS implements a scheme to conduct the integrity check for the executable code of certain modules during its initial startup. The scheme starts with the Windows OS boot manager (bootmgr.efi on EFI firmware machines or bootmgr.exe on PC/AT firmware machines), after a user physically pushes the power button of the hardware machine, on which the Windows OS resides. The Windows OS boot manager conducts a self integrity check before it attempts to load other Windows OS modules.

### Integrity check conducted by the Windows OS boot manager

The Windows OS boot manager is an [Authenticode signed image file](http://www.microsoft.com/whdc/winlogo/drvsign/Authenticode_PE.mspx), based on the [Portable Executable (PE) image file](http://msdn.microsoft.com/en-us/library/ms680336.aspx) format. A SHA-1 hash based signature and a public key certificate chain are embedded in the Windows OS boot manager [Authenticode signed image file](http://www.microsoft.com/whdc/winlogo/drvsign/Authenticode_PE.mspx) under the “Certificate” IMAGE\_DATA\_DIRECTORY of the [IMAGE\_OPTIONAL\_HEADER](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) of the file. This public key certificate chain ends in a root public key. The Windows OS boot manager uses the embedded SHA-1 hash based signature and public key certificate chain to validate its own integrity. A SHA-1 hash of the Windows OS boot manager image file is calculated for the whole file, except that the following three elements are excluded from the hash calculation:

* the CheckSum field in the IMAGE\_OPTIONAL\_HEADER;
* the [IMAGE\_DIRECTORY\_ENTRY\_SECURITY](http://msdn.microsoft.com/en-us/library/ms680149.aspx) IMAGE\_DATA\_DIRECTORY;
* the public key certificate table, which always resides at the end of the image file.

If the Windows OS boot manager is validated, then the root public key of the embedded public key certificate chain must match one of the Microsoft’s root public keys which indicate that Microsoft is the publisher of the Windows OS boot manager. These Microsoft’s root public keys are necessarily hardcoded in the Windows OS boot manager. If the Windows OS boot manager cannot validate its own integrity, then the Windows OS boot manager does not continue to load other Windows OS modules and displays the following error message on the screen.

“Windows cannot verify the digital signature for this file <indicating the Windows OS boot manager (bootmgr.efi or bootmgr.exe)>.

A recent hardware or software change might have installed a file that is signed incorrectly or damaged, or that might be malicious software from an unknown source.

If you have a Windows installation disc, insert the disc and restart your computer. Click "Repair your computer," and then choose a recovery tool.

Otherwise, to start Windows so you can investigate further, press the ENTER key to display the boot menu, press F8 for Advanced Boot Options, and select Last Known Good. If you understand why the digital signature cannot be verified and want to start Windows without this file, temporarily disable driver signature enforcement.”

The Windows OS boot manager is a part of the Windows OS BitLocker™ components that have received FIPS-140-2 (Cert # 947 and Cert # 1054).

After the Windows OS boot manager determinates its own integrity, it proceeds to attempt to load one of the following Windows OS boot applications, as described earlier in the “Validation information and the encrypted volume encryption key” section of this paper.

* winload.exe (on PC/AT firmware machines) or winload.efi (on EFI firmware machines), which is the application to load the Windows OS, namely ntoskrnl.exe;
* winresume.exe (on PC/AT firmware machines) or winresume.efi (on EFI firmware machines), which is the application for resuming the Windows OS from the hibernation file “hiberfil.sys”;
* memtest.exe (on PC/AT firmware machines) or memtest.efi (on EFI firmware machines), which is a memory testing application.

These Windows OS boot applications are also an [Authenticode signed image file](http://www.microsoft.com/whdc/winlogo/drvsign/Authenticode_PE.mspx), based on the [Portable Executable (PE) image file](http://msdn.microsoft.com/en-us/library/ms680336.aspx) format. For each of these Windows OS boot applications, the Windows OS boot manager uses the embedded trusted SHA-1 hash based signature and public key certificate chain within the Windows OS boot application’s [IMAGE\_OPTIONAL\_HEADER](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) to validate the integrity of the Windows OS boot application before attempting to load it. A SHA-1 hash of a Windows OS boot application image file is calculated for the whole file, except that the following three elements are excluded from the hash calculation:

* the CheckSum field in the IMAGE\_OPTIONAL\_HEADER;
* the [IMAGE\_DIRECTORY\_ENTRY\_SECURITY](http://msdn.microsoft.com/en-us/library/ms680149.aspx) IMAGE\_DATA\_DIRECTORY;
* the public key certificate table, which always resides at the end of the image file.

If the Windows OS boot application is validated, then the root public key of the embedded public key certificate chain must match one of the Windows OS boot manager’s hardcoded Microsoft’s root public keys which indicate that Microsoft is the publisher of the application. If the Windows OS boot manager cannot validate the integrity of a Windows OS boot application, then the Windows OS boot manager does not continue to load other Windows OS modules and displays the following error message, with an indication of the full name of the Windows OS boot application that has failed the integrity check, on the screen.

“Windows cannot verify the digital signature for this file.

A recent hardware or software change might have installed a file that is signed incorrectly or damaged, or that might be malicious software from an unknown source.

If you have a Windows installation disc, insert the disc and restart your computer. Click "Repair your computer," and then choose a recovery tool.

Otherwise, to start Windows so you can investigate further, press the ENTER key to display the boot menu, press F8 for Advanced Boot Options, and select Last Known Good. If you understand why the digital signature cannot be verified and want to start Windows without this file, temporarily disable driver signature enforcement.”

After the Windows OS boot manager determinates the Windows OS boot application’s integrity, it proceeds to attempt to load the Windows OS boot application. As mentioned in the “Validation information and the encrypted volume encryption key” section of this paper, the full volume encryption (FVE) facility within the Windows OS boot manager also conducts its own independent SHA256 hash based validation of the above Windows OS boot applications.

If the Windows OS boot application is successfully loaded, the Windows OS boot manager transfers the execution to the loaded Windows OS boot application.

The Windows OS winload boot application is a part of the Windows OS BitLocker™ components that have received FIPS-140-2 (Cert # 947 and Cert # 1054).

The Windows OS winresume boot application is a part of the Windows OS BitLocker™ components that have received FIPS-140-2 (Cert # 947 and Cert # 1054).

### Integrity check conducted by the Windows OS boot winload application

After the Windows OS boot winload application is loaded, it receives the transfer of execution from the Windows OS boot manager. During its execution, the Windows OS winload boot application attempts to load the Windows OS kernel, namely ntoskrnl.exe, together with a number of critical drivers. Among the others, the Windows OS winload boot application must validate the following cryptography related Windows OS modules, in the [Portable Executable (PE) image file](http://msdn.microsoft.com/en-us/library/ms680336.aspx) format, before it attempts to load them into the memory. They are listed in the hardcoded “Microsoft Boot Images” list residing inside the Windows OS boot winload application.

* The Windows OS kernel (ntoskrnl.exe);
* The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys)
  + It has received FIPS-140-2 (Cert # 891 and Cert # 1007);
* The Windows OS code integrity library module (ci.dll)
  + It is a part of the Windows OS BitLocker™ components that have received FIPS-140-2 (Cert # 947 and Cert # 1054);
* The BitLocker™ drive encryption filter driver (fvevol.sys)
  + It is a part of the Windows OS BitLocker™ components that have FIPS-140-2 (Cert # 947 and Cert # 1054).

The above four Windows OS image files have their trusted SHA-1 hashes stored in the Microsoft’s [catalog](http://msdn.microsoft.com/en-us/library/aa906236.aspx) files. These [catalog](http://msdn.microsoft.com/en-us/library/aa906236.aspx) files reside in the local machine catalog directory (namely “$SystemRoot$\System32\CatRoot\{F750E6C3-38EE-11D1-85E5-00C04FC295EE}”). As a PKCS #7 SignedData message, these [catalog](http://msdn.microsoft.com/en-us/library/aa906236.aspx) files are necessarily signed. The root public key of the certificate chain used to verify the signature of a Microsoft’s catalog file must match one of the Microsoft’s root public keys which indicate that Microsoft is the publisher of the aforementioned Windows OS image files. These Microsoft’s root public keys are necessarily hardcoded in the Windows OS winload boot application.

If the aforementioned Windows OS image files are validated, their SHA-1 hashes, as calculated by the Windows OS winload boot application, must match their trusted SHA-1 hashes in a Microsoft’s [catalog](http://msdn.microsoft.com/en-us/library/aa906236.aspx) file, which has been verified by the Windows OS winload boot application. A SHA-1 hash of a Windows OS image file is calculated for the whole file, except that the following three elements are excluded from the hash calculation:

* the CheckSum field in the IMAGE\_OPTIONAL\_HEADER;
* the [IMAGE\_DIRECTORY\_ENTRY\_SECURITY](http://msdn.microsoft.com/en-us/library/ms680149.aspx) IMAGE\_DATA\_DIRECTORY;
* the public key certificate table, which always resides at the end of the image file.

If the Windows OS winload boot application cannot validate the integrity of one of the aforementioned Windows OS image files, then the Windows OS winload boot application does not continue to load other Windows OS image files and displays the following error message, with an indication of the full name of the Windows OS image file which does not have the validated integrity, on the screen.

“Windows failed to load because the kernel is missing, or corrupt [in the case where the integrity of ntoskrnl.exe is not validated], or

Windows failed to load because a critical system driver is missing, or corrupt [in the case where the integrity of ksecdd.sys, ci.dll, fvevol.sys, or other critical driver is not validated].

Windows cannot verify the digital signature for this file.”

After the Windows OS winload boot application determinates the integrity of the Windows OS kernel (ntoskrnl.exe) and the Windows OS critical drivers, it proceeds to attempt to load them into the memory.

If the Windows OS kernel (ntoskrnl.exe) is successfully loaded, the Windows OS winload boot application transfers the execution to the loaded Windows OS kernel (ntoskrnl.exe) by jumping to the Windows OS kernel’s entry point.

After the Windows OS kernel (ntoskrnl.exe) is loaded, it receives the transfer of execution from the Windows OS winload boot application. Having received the execution, the Windows OS kernel goes through all the initialization steps to get all the integrated Windows OS modules and the loaded critical drivers ready for operations.

The Windows OS security reference monitor (se.lib) is an integrated Windows OS module of the Windows OS kernel. As explained in the “Implementation locations of the Windows OS access determination algorithm” section of this paper, the Windows OS security reference monitor (se.lib) contains one of the two locations that implement the “Windows OS access determination algorithm”.

### Integrity check conducted by the Windows OS code integrity library module

The Windows OS memory manager (mm.lib) is also an integrated Windows OS module of the Windows OS kernel. The Windows OS memory manager is in the critical path when the Windows OS memory manager is requested to create a [memory section object](http://msdn.microsoft.com/en-us/library/ms796304.aspx) by reading a specified Windows OS image file and then mapping the read file to the [memory section object](http://msdn.microsoft.com/en-us/library/ms796304.aspx) being created. If the specified Windows OS image file is one of the following types, then the Windows OS memory manager uses the capability of the already loaded Windows OS code integrity library module (ci.dll) to determine the integrity of the Windows OS image file, before attempting to map the Windows OS image file.

* The Windows OS image file is an executable which is expected to run in the [protected environment as a protected process](http://msdn.microsoft.com/en-us/library/aa376846.aspx) (SE\_VALIDATE\_PROTECTED\_IMAGE/CI\_VALIDATE\_PROTECTED\_IMAGE);
* The [memory section object](http://msdn.microsoft.com/en-us/library/ms796304.aspx) creation request to map the Windows OS image file comes from a [protected process running in the protected environment](http://msdn.microsoft.com/en-us/library/aa376846.aspx) (SE\_VALIDATE\_PROTECTED\_IMAGE/CI\_VALIDATE\_PROTECTED\_IMAGE);
* The Windows OS image file is a kernel mode driver (SE\_VALIDATE\_DRIVER\_IMAGE/CI\_VALIDATE\_DRIVER\_IMAGE);
* The Windows OS image file is a kernel mode driver in a hot-patching scenario (SE\_VALIDATE\_HOTPATCH\_DRIVER\_IMAGE/CI\_VALIDATE\_HOTPATCH\_DRIVER\_IMAGE);
* The [IMAGE\_DLLCHARACTERISTICS\_FORCE\_INTEGRITY](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) flag is set in the [DllCharacteristics](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) field of the [IMAGE\_OPTIONAL\_HEADER](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) of the Windows OS image file (CI\_VALIDATE\_CRYPTO\_IMAGE), requiring that the code integrity check for the Windows OS image file is enforced.

As a result, the following Windows OS modules, which are relevant in this Commercial Grade OS Requirement Set “5.3.1.3” requirement, are subject to the integrity check as the Windows OS memory manager is requested to attempt to map one of them into a [memory section object](http://msdn.microsoft.com/en-us/library/ms796304.aspx) as a Windows OS image file.

* DUMPFVE.SYS
  + It is a Windows OS BitLocker™ component for supporting the Full volume encryption. It is a part of the Windows OS BitLocker™ components that have received FIPS-140-2 (Cert # 947 and Cert # 1054). As a BitLocker™ filter driver, it sits in the system dump stack. Whenever the dump stack is called (in the event of a crash, or for hibernation), this filter ensures that all data is encrypted before it is written to the disk (in a dump file or hibernation file). This component is subject to the integrity check because it is a kernel mode driver (SE\_VALIDATE\_DRIVER\_IMAGE/CI\_VALIDATE\_DRIVER\_IMAGE).
* FVEAPI.DLL
  + It is a Windows OS BitLocker™ component for supporting the Full volume encryption. It is a part of the Windows OS BitLocker™ components that have received FIPS-140-2 (Cert # 947 and Cert # 1054). As a user mode DLL library, it provides library functions for the BitLocker™ key generation and key management. This component is subject to the integrity check because the [IMAGE\_DLLCHARACTERISTICS\_FORCE\_INTEGRITY](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) flag is set in the [DllCharacteristics](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) field of its [IMAGE\_OPTIONAL\_HEADER](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx).
* WIN32\_TPM.DLL
  + It is a Windows OS BitLocker™ component for supporting the Full volume encryption. It is a part of the Windows OS BitLocker™ components that have received FIPS-140-2 (Cert # 947 and Cert # 1054). As a user mode [Windows Management Instrumentation (WMI) provider DLL for the TPM API](http://msdn.microsoft.com/en-us/library/aa376484.aspx), it provides library functions for controlling the TPM functionality. This component is subject to the integrity check because the [IMAGE\_DLLCHARACTERISTICS\_FORCE\_INTEGRITY](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) flag is set in the [DllCharacteristics](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) field of its [IMAGE\_OPTIONAL\_HEADER](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx).
* bcrypt.dll
  + It is the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx). It has received FIPS-140-2 (Cert # 892 and Cert # 1008). This component is subject to the integrity check because the [IMAGE\_DLLCHARACTERISTICS\_FORCE\_INTEGRITY](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) flag is set in the [DllCharacteristics](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) field of its [IMAGE\_OPTIONAL\_HEADER](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx).
* ncrypt.dll
  + It is the wrapper for the FIPS 140-2 validated [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) to support the long lived private key storage and retrieval in the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) when using the public key cryptographic services of the [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll). This component is subject to the integrity check because the [IMAGE\_DLLCHARACTERISTICS\_FORCE\_INTEGRITY](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) flag is set in the [DllCharacteristics](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) field of its [IMAGE\_OPTIONAL\_HEADER](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx).
* rsaenh.dll
  + It is the Windows OS [Enhanced Cryptographic Provider](http://msdn.microsoft.com/en-us/library/aa386979(VS.85).aspx). It has received FIPS-140-2 (Cert # 893 and Cert # 1010). This component is subject to the integrity check because the [IMAGE\_DLLCHARACTERISTICS\_FORCE\_INTEGRITY](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) flag is set in the [DllCharacteristics](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) field of its [IMAGE\_OPTIONAL\_HEADER](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx).
* dssenh.dll
  + It is the [Windows OS Enhanced DSS and Diffie-Hellman Cryptographic Provider](http://msdn.microsoft.com/en-us/library/bb394802(VS.85).aspx). It has received FIPS-140-2 (Cert # 894 and Cert # 1009). This component is subject to the integrity check because the [IMAGE\_DLLCHARACTERISTICS\_FORCE\_INTEGRITY](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) flag is set in the [DllCharacteristics](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) field of its [IMAGE\_OPTIONAL\_HEADER](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx).

In the case where the Windows OS module, being checked for its integrity as a Windows OS image file, is a kernel mode driver (such as DUMPFVE.SYS), the Windows OS image file has its trusted SHA-1 hash stored in the Microsoft’s [catalog](http://msdn.microsoft.com/en-us/library/aa906236.aspx) files. These [catalog](http://msdn.microsoft.com/en-us/library/aa906236.aspx) files reside in the local machine catalog directory (namely “$SystemRoot$\System32\CatRoot\{F750E6C3-38EE-11D1-85E5-00C04FC295EE}”). As a PKCS #7 SignedData message, these [catalog](http://msdn.microsoft.com/en-us/library/aa906236.aspx) files are necessarily signed. The root public key of the certificate chain used to verify the signature of a Microsoft’s catalog file must match one of the Microsoft’s root public keys which indicate that Microsoft is the publisher of the Windows OS image file. These Microsoft’s root public keys are necessarily hardcoded in the Windows OS code integrity library module (ci.dll).

If the Windows OS image file is validated, its SHA-1 hash, as calculated by the Windows OS code integrity library module (ci.dll), must match its trusted SHA-1 hash in a Microsoft’s [catalog](http://msdn.microsoft.com/en-us/library/aa906236.aspx) file, which has been verified by the Windows OS code integrity library module (ci.dll). A SHA-1 hash of a Windows OS image file is calculated for the whole file, except that the following three elements are excluded from the hash calculation:

* the CheckSum field in the IMAGE\_OPTIONAL\_HEADER;
* the [IMAGE\_DIRECTORY\_ENTRY\_SECURITY](http://msdn.microsoft.com/en-us/library/ms680149.aspx) IMAGE\_DATA\_DIRECTORY;
* the public key certificate table, which always resides at the end of the image file.

If the Windows OS code integrity library module (ci.dll) cannot validate the integrity of the Windows OS image file, then the Windows OS code integrity library module (ci.dll) returns the [STATUS\_INVALID\_IMAGE\_HASH](http://msdn.microsoft.com/en-us/library/cc704588.aspx) error code to the Windows OS memory manager. The Windows OS memory manager does not continue the attempt to map the Windows OS image file into the requested [memory section object](http://msdn.microsoft.com/en-us/library/ms796304.aspx). The Windows OS code integrity library module (ci.dll) also generates the following audit event record to indicate the specific name of the Windows OS image file that has failed the integrity check.

* [Event ID 5038](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_CODE\_INTEGRITY\_INVALID\_IMAGE\_HASH\_value) security audit record “Code integrity determined that the image hash of a file is not valid. The file could be corrupt due to unauthorized modification or the invalid hash could indicate a potential disk device error”.

In the case where the Windows OS module, being checked for its integrity as a Windows OS image file, is a DLL (such as FVEAPI.DLL, WIN32\_TPM.DLL, bcrypt.dll, rsaenh.dll, or dssenh.dll), the Windows OS image file has its full set of trusted SHA-1 page hashes stored in the Microsoft’s [catalog](http://msdn.microsoft.com/en-us/library/aa906236.aspx) files.

In the full set of SHA-1 page hashes of a Windows OS image file, the first SHA-1 hash is the SHA-1 hash of the [image header](http://msdn.microsoft.com/en-us/library/ms680336(VS.85).aspx) of the Windows OS image file, and each of the subsequent SHA-1 hashes is calculated over a page of the rest of the Windows OS image file at a time for all the pages of the Windows OS image file after the image header. Three elements in the Windows OS image file are excluded in the hash calculation. They are:

* the CheckSum field in the IMAGE\_OPTIONAL\_HEADER;
* the [IMAGE\_DIRECTORY\_ENTRY\_SECURITY](http://msdn.microsoft.com/en-us/library/ms680149.aspx) IMAGE\_DATA\_DIRECTORY;
* the public key certificate table, which always resides at the end of the image file.

On IA64 machine, the page size is 8K. On other machines (including x86 and x64), the page size is 4K.

The [catalog](http://msdn.microsoft.com/en-us/library/aa906236.aspx) files, storing the full sets of trusted SHA-1 page hashes, reside in the local machine catalog directory (namely “$SystemRoot$\System32\CatRoot\{F750E6C3-38EE-11D1-85E5-00C04FC295EE}”). As a PKCS #7 SignedData message, these [catalog](http://msdn.microsoft.com/en-us/library/aa906236.aspx) files are necessarily signed. The root public key of the certificate chain used to verify the signature of a Microsoft’s catalog file must match one of the Microsoft’s root public keys which indicate that Microsoft is the publisher of the Windows OS image files. These Microsoft’s root public keys are necessarily hardcoded in the Windows OS code integrity library module (ci.dll).

If the Windows OS image file is validated, its full set of SHA-1 page hashes, as calculated by the Windows OS code integrity library module (ci.dll), must match its full set of trusted SHA-1 page hashes in a Microsoft’s [catalog](http://msdn.microsoft.com/en-us/library/aa906236.aspx) file, which has been verified by the Windows OS code integrity library module (ci.dll).

It is recommended that an authorized administrator shall manually inspect that the [IMAGE\_DLLCHARACTERISTICS\_FORCE\_INTEGRITY](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) flag is set in the [DllCharacteristics](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) field of the [IMAGE\_OPTIONAL\_HEADER](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) of each of the following DLLs, which are relevant for addressing this Commercial Grade OS Requirement Set “5.3.1.3” requirement:

* FVEAPI.DLL;
* WIN32\_TPM.DLL;
* bcrypt.dll;
* ncrypt.dll;
* rsaenh.dll;
* dssenh.dll.

If the Windows OS code integrity library module (ci.dll) cannot validate the integrity of the Windows OS image file, then the Windows OS code integrity library module (ci.dll) returns the [STATUS\_INVALID\_IMAGE\_HASH](http://msdn.microsoft.com/en-us/library/cc704588.aspx) error code to the Windows OS memory manager. The Windows OS memory manager does not continue the attempt to map the Windows OS image file into the requested [memory section object](http://msdn.microsoft.com/en-us/library/ms796304.aspx). The Windows OS code integrity library module (ci.dll) also generates the following audit event record to indicate the specific name of the Windows OS image file that has failed the integrity check.

* [Event ID 5038](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_CODE\_INTEGRITY\_INVALID\_IMAGE\_HASH\_value) security audit record “Code integrity determined that the image hash of a file is not valid. The file could be corrupt due to unauthorized modification or the invalid hash could indicate a potential disk device error”.

### Chaining together the integrity checks

In summary, the above have described the integrity check of the executable code of the following Windows OS modules that implement access control and cryptographic functionality through the use of the Windows OS system provided cryptographic services.

* The Windows OS boot manager (bootmgr.efi or bootmgr.exe)
  + it is critical to conduct the integrity check of itself;
  + it is critical to conduct the integrity check of and to load the Windows OS boot applications;
* The Windows OS boot winload application (winload.efi or winload.exe)
  + it is critical to conduct the integrity check of and to load the Windows OS kernel and other critical drivers;
* The Windows OS boot winresume application (winresume.efi or winresume.exe) as a part of the Windows OS BitLocker™ components
  + It is critical to resume the Windows OS from the hibernation file “hiberfil.sys”;
* The Windows OS kernel (ntoskrnl.exe)
  + It includes the Windows OS security reference monitor (se.lib), which is critical for conducting the “Windows OS access determination algorithm” in the kernel mode;
  + It includes the Windows OS memory manager (mm.lib), which is critical to use the Windows OS code integrity library module (ci.dll) to conduct the integrity check before attempting to map a specified Windows OS image file into a requested [memory section object](http://msdn.microsoft.com/en-us/library/ms796304.aspx);
* The [Windows OS kernel security device driver](http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140sp/140sp1007.pdf) (ksecdd.sys);
* The Windows OS code integrity library module (ci.dll);
* The BitLocker™ drive encryption filter driver (fvevol.sys);
* The Windows OS DUMPFVE.SYS BitLocker™ component for supporting the Full volume encryption;
* The Windows OS FVEAPI.DLL BitLocker™ component for supporting the Full volume encryption;
* The Windows OS WIN32\_TPM.DLL BitLocker™ component for supporting the Full volume encryption;
* The Windows OS Cryptographic Primitives Library (bcrypt.dll);
* The wrapper (ncrypt.dll) for the Windows OS Cryptographic Primitives Library (bcrypt.dll) to support the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll);
* The Windows OS Enhanced Cryptographic Provider (rsaenh.dll);
* The Windows OS Enhanced DSS and Diffie-Hellman Cryptographic Provider (dssenh.dll).

As explained in the “Implementation locations of the Windows OS access determination algorithm” section of this paper, the user mode authorization framework library (authz.dll) is the other Windows OS module that implements the “Windows OS access determination algorithm”. Unfortunately, the [IMAGE\_DLLCHARACTERISTICS\_FORCE\_INTEGRITY](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) flag is not set in the [DllCharacteristics](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) field of the [IMAGE\_OPTIONAL\_HEADER](http://msdn.microsoft.com/en-us/library/ms680339(VS.85).aspx) of authz.dll as a Windows OS image file. Therefore, its integrity check does not occur as the Windows OS memory manager (mm.lib) attempts to map it into a requested [memory section object](http://msdn.microsoft.com/en-us/library/ms796304.aspx).

As a result, except for the case of the user mode authorization framework library (authz.dll), this Commercial Grade OS Requirement Set “5.3.1.3” requirement is met.

Finally it should be mentioned that, in a running system after the proper installation of Windows OS, the Windows OS system executable files such as authz.dll reside in the “$SystemRoot$\system32” directory of the local file system. These files have the default file object security descriptor, and the directory has the default file directory object security descriptor as follows.

* The default file object security descriptor has the Windows OS Trusted Installer (S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464) as the owner. It grants the built-in “Administrators” group, the local system, and the built-in “Users” group only the generic read and execute access rights.
* The default file directory object security descriptor has the Windows OS Trusted Installer (S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464) as the owner. It grants the built-in “Administrators” group and the local system only the modify access rights, and it grants the built-in “Users” group only the generic read and execute access rights.

# Meeting the “Information Protection Trusted Initialization~~, Self Testing and Recovery~~ Management Requirements”

~~In the Commercial Grade OS Requirement Set, there are 2 individual management requirements under the heading of “Information Protection Trusted Initialization, Self Testing and Recovery Management Requirements”. They are listed as “5.3.2.n”, where n = 1 and 2.~~ In the Commercial Grade OS Requirement Set, there is no individual management requirement under the heading of “Information Protection Trusted Initialization Management Requirements”.

## ~~Addressing 5.3.2.1 “The OS shall provide authorized administrators the ability to run the OS self tests”~~

~~This requirement is moot because the Windows OS itself does not incorporate self tests for its security functions specified in this Commercial Grade OS Requirement Set as parts of its operational features.~~

## ~~Addressing 5.3.2.2 “The OS shall provide authorized administrators the ability to define the frequency at which self tests are automatically run”~~

~~This requirement is moot because the Windows OS itself does not incorporate self tests for its security functions specified in this Commercial Grade OS Requirement Set as parts of its operational features.~~

# Meeting the “Information Protection Trusted Initialization~~, Self Testing and Recovery~~ Audit Requirements”

In the Commercial Grade OS Requirement Set, there ~~are 3 individual audit requirements~~ is 1 individual audit requirement under the heading of “Information Protection Trusted Initialization~~, Self Testing and Recovery~~ Audit Requirements”. ~~They are listed as “5.3.3.n”, where n = 1, 2 and 3.~~ It is listed as “5.3.3.3”.

## ~~Addressing 5.3.3.1 “The OS shall provide the ability to audit any self test failures”~~

~~This requirement is moot because the Windows OS itself does not incorporate self tests for its security functions specified in this Commercial Grade OS Requirement Set as parts of its operational features.~~

## ~~Addressing 5.3.3.2 “The OS shall provide the ability to audit when authorized administrators initiate the OS self tests”~~

~~This requirement is moot because the Windows OS itself does not incorporate self tests for its security functions specified in this Commercial Grade OS Requirement Set as parts of its operational features.~~

## Addressing 5.3.3.3 “The OS shall provide the ability to audit the failure of the (executable code) integrity verification”

This requirement is addressed by the Windows OS as follows.

Recalling from the justification text for addressing the Commercial Grade OS Requirement Set “5.3.1.3” requirement, the Windows OS code integrity library module (ci.dll) generates the following audit event record to indicate the specific name of the Windows OS image file that has failed the integrity check.

* [Event ID 5038](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_CODE\_INTEGRITY\_INVALID\_IMAGE\_HASH\_value) security audit record “Code integrity determined that the image hash of a file is not valid. The file could be corrupt due to unauthorized modification or the invalid hash could indicate a potential disk device error”.

Before the Windows OS kernel (ntoskrnl.exe) is loaded and initialized, if the Windows OS boot manager cannot validate the integrity of itself or a Windows OS boot application, then the Windows OS boot manager does not continue to load other Windows OS modules and displays the following error message, with an indication of the full name of itself or the Windows OS boot application that has failed the integrity check, on the screen.

“Windows cannot verify the digital signature for this file.

A recent hardware or software change might have installed a file that is signed incorrectly or damaged, or that might be malicious software from an unknown source.

If you have a Windows installation disc, insert the disc and restart your computer. Click "Repair your computer," and then choose a recovery tool.

Otherwise, to start Windows so you can investigate further, press the ENTER key to display the boot menu, press F8 for Advanced Boot Options, and select Last Known Good. If you understand why the digital signature cannot be verified and want to start Windows without this file, temporarily disable driver signature enforcement.”

In addition, if the Windows OS winload boot application cannot validate the integrity of any of the critical Windows OS image files in the hardcoded “Microsoft Boot Images” list for loading into the memory, then the Windows OS winload boot application does not continue to load other Windows OS image files and displays the following error message, with an indication of the full name of the Windows OS image file which does not have the validated integrity, on the screen.

“Windows failed to load because the kernel is missing, or corrupt [in the case where the integrity of ntoskrnl.exe is not validated], or

Windows failed to load because a critical system driver is missing, or corrupt [in the case where the integrity of ksecdd.sys, ci.dll, fvevol.sys, or other critical driver is not validated].

Windows cannot verify the digital signature for this file.”

As a result, this Commercial Grade OS Requirement Set “5.3.3.3” requirement is met.

# Meeting the “Import/Export of Data Exported Data Functional Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual functional requirement under the heading of “Import/Export of Data Exported Data Functional Requirements”. It is listed as “6.1.1.1”.

## Addressing 6.1.1.1 “The OS shall ensure that security attributes on named objects, when exported to removable media, are associated with the object”

This requirement is addressed by the Windows OS as follows.

As explained in the “Encrypting file system (EFS)” section, a location where the encryption/decryption of EFS-protected files occurs is the [SMB](http://msdn2.microsoft.com/en-us/library/cc246231.aspx) remote file system redirector (mrxsmb20.sys). A subject needs to use the kernel mode interfaces to the Windows OS IO manager in order to gain access to the services of the redirector. The [SMB](http://msdn2.microsoft.com/en-us/library/cc246231.aspx) remote file system redirector supports the write and read access to named files residing remotely across the network. In this SMB redirector case, file encryption and decryption occurs locally. Therefore, the redirector sends encrypted file data of a specific named file over SMB to a remote server for storage, and it receives encrypted file data of the specific named file from a remote server over SMB. The remote server processes the encrypted file data as raw file data only for storage. The remote server need not be a Windows OS machine, but any machine (including 3rd party machines) that supports the [SMB](http://msdn2.microsoft.com/en-us/library/cc246231.aspx) protocol. By a 3rd party machine, we mean a machine that is not running the Windows OS, but a certain 3rd party operating system. Consequently, the sending of the FSCTL\_LMR\_EFS\_WRITE\_RAW FSCTL over the [SMB](http://msdn2.microsoft.com/en-us/library/cc246231.aspx) protocol from the [SMB](http://msdn2.microsoft.com/en-us/library/cc246231.aspx) remote file system redirector to the remote server is deemed as data exporting to a removable media. Obviously, the data being exported in this case is the EFS-encrypted file data.

The EFS-encrypted file data includes exactly one “$EFS” metadata stream. The “$EFS” metadata stream contains

* one or more [Data Decryption Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) (DataDecryptionField);
* zero or more [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) (DataRecoveryField).

There is at least one Data Decryption Field, which contains the encrypted File Encryption Key (FEK).

Subject to the Windows OS domain wide EFS policy, the FEK is RSA-encrypted with the EFS certificate public key of the user account belonging to the user subject who creates the specific named file. By default, the size of the EFS certificate public key is 2048 bits. The Windows OS domain wide EFS policy also allows an option where the FEK is AES-encrypted with an AES 256 master key. This AES 256 master key is protected by the local [Windows OS data protection API (DPAPI) manager](http://msdn.microsoft.com/en-us/library/ms995355.aspx) for the user account belonging to the user subject who creates the specific named file. The other Data Decryption Fields (if they exist) also contain a copy of the encrypted FEK. Each copy of the FEK is RSA-encrypted with the EFS certificate public key of a user account which has been authorized to have access to the clear-text data of the specific named file. The authorization may be granted through the [EfsRpcAddUsersToFile()](http://msdn.microsoft.com/en-us/library/cc230472.aspx) or [EfsRpcAddUsersToFileEx()](http://msdn.microsoft.com/en-us/library/cc230473.aspx) interfaces of the EFS key management unit residing within the Windows OS authentication service process (lsass.exe).

Depending on the Windows OS domain wide [EFS policy](http://msdn.microsoft.com/en-us/library/cc232262.aspx), there could be a Data Recovery Field, which also contains the encrypted File Encryption Key (FEK). The FEK is RSA-encrypted with the EFS Recovery Agent certificate public key of an EFS Recovery Agent specified in the [EFS policy](http://msdn.microsoft.com/en-us/library/cc232262.aspx). By default, the size of the EFS Recovery Agent certificate public key is 2048 bits.

When configured, the [EFS policy](http://msdn.microsoft.com/en-us/library/cc232262.aspx) is distributed to individual Windows OS machines, belonging to the Windows OS domain/forest, through the [Windows OS group policy](http://msdn2.microsoft.com/en-us/library/aa374177.aspx).

From the perspective of the clear-text data of the specific named file, the [Data Decryption Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) and the [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) are deemed as its security attributes. The association between the clear-text data and its [Data Decryption Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) and [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) security attributes is ensured by the File Encryption Key (FEK). While the clear-text data is encrypted with the FEK, the FEK is encrypted with either

* a DPAPI-protected AES 256 master key;
* an EFS certificate public key;
* an EFS Recovery Agent certificate public key.

Given the EFS-encrypted file data of the specific named file, if the access to a valid FEK from the security attributes is unsuccessful, then so is the access to the corresponding clear-text data.

The local EFS encryption and decryption is also supported in the Windows OS for named files being exported over the [Web Distributed Authoring and Versioning (WebDAV) protocol (RFC 2518)](http://msdn.microsoft.com/en-us/library/aa364664.aspx) to a remote WebDAV server. Similarly, the remote WebDAV server need not be a Windows OS machine, but any machine (including 3rd party machines) that supports the WebDAV protocol.

Therefore, both SMB and WebDAV are the transport protocols for exporting the EFS-encrypted file data of named files with their [Data Decryption Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) and [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) security attributes to remote servers for storage.

Finally, we note that, while residing on a remote server, the EFS-encrypted file data is just storage data. As storage data, the EFS-encrypted file data subsequently can be copied, by anyone who has physical access to the remote server, to a removable media as wished.

As a result, this Commercial Grade OS Requirement Set “6.1.1.1” requirement is met.

# Meeting the “Import/Export of Data Exported Data Management Requirements”

In the Commercial Grade OS Requirement Set, there is no individual management requirement under the heading of “Import/Export of Data Exported Data Management Requirements”.

# Meeting the “Import/Export of Data Exported Data Audit Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual audit requirement under the heading of “Import/Export of Data Exported Data Audit Requirements”. It is listed as “6.1.3.1”.

## Addressing 6.1.3.1 “The OS shall provide the ability to audit the exportation of named objects to removable media”

This requirement is addressed by the Windows OS as follows.

In the scenario described in the above for exporting certain EFS-encrypted file data to a remote SMB or WebDAV server, the subject actually interacts with a specific named file which is residing in the remote server and not in the local Windows OS machine where the subject resides in. The subject merely writes the clear-text data into the specific named file, but having the local Encrypting file system (EFS) to encrypt the clear-text data into the EFS-encrypted file data locally before the EFS-encrypted file data is transported over SMB or WebDAV to the remote server. The remote server then writes the transported EFS-encrypted file data into the specific named file as raw file data only for storage. This scenario is similar to the scenario where the subject interacts with a specific named web server over a [HTTPS](http://msdn.microsoft.com/en-us/library/aa767735(VS.85).aspx) connection and the arbitrary contents entered by the subject are encrypted by the [Schannel security provider](http://msdn.microsoft.com/en-us/library/aa380123(VS.85).aspx) which handles the underlying [RFC 2246 Transport Layer Security (TLS)](http://msdn2.microsoft.com/en-us/library/aa380123(VS.85).aspx) protocol processing for the [HTTPS](http://msdn.microsoft.com/en-us/library/aa767735(VS.85).aspx) connection.

In both scenarios, the local generation of an audit record for the event where the subject is writing arbitrary contents to a specific named file residing in a remote server seems unwarranted. If the remote server were a distributed portion of the same Windows OS as the local Windows OS machine, then there would be the ability for generating the following audit event records, according to the justification text for addressing the Commercial Grade OS Requirement Set “2.1.3.1” requirement, at the remote server, to indicate the access to the specific named file residing in the remote server.

* [Event ID 4656](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OPEN\_HANDLE\_value) “A handle to an object was requested”;
* [Event ID 4663](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OBJECT\_ACCESS\_value) “An attempt was made to access an object”.

The contents written by the subject to the specific named file residing in the remote server actually come from either

* the direct editing of the user represented by the subject;
* another source named file residing in the local Windows OS machine through a means such as “copy and paste” or “save as” of an application of the subject;
* another source named file residing in another source remote Windows OS machine through a means such as “copy and paste” or “save as” of an application of the subject.

In the “direct editing” case, the local audit generation is unwarranted as there is also no warranted audit generation requirement when the subject edits contents directly during the interaction with a specific named web server over a [HTTPS](http://msdn.microsoft.com/en-us/library/aa767735(VS.85).aspx) connection.

In the “from another source named file residing in the local Windows OS machine” case, the subject would have opened the source named file for the read access successfully. Therefore, the audit generation of [Event ID 4656](http://support.microsoft.com/kb/947226) and [Event ID 4663](http://support.microsoft.com/kb/947226) would have occurred in the local Windows OS machine to indicate the read access to the source named file.

In the “from another source named file residing in another source remote Windows OS machine” case, the subject would have opened the source named file for the read access successfully. Therefore, the audit generation of [Event ID 4656](http://support.microsoft.com/kb/947226) and [Event ID 4663](http://support.microsoft.com/kb/947226) would have occurred in the source remote Windows OS machine to indicate the read access to the source named file.

It should also be noted that, in general, after the subject has been granted the read access to a specific named file (which is audited subject to the audit policy), it is not the Windows OS responsibility to control or audit what the subject would do with any content that the subject may read from the named file.

Given the above explanations, this Commercial Grade OS Requirement Set “6.1.3.1” requirement is addressed.

# Meeting the “Import/Export of Data Imported Data Functional Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual functional requirement under the heading of “Import/Export of Data Imported Data Functional Requirements”. It is listed as “6.2.1.1”.

## Addressing 6.2.1.1 “The OS shall enforce a specific set of rules when importing user data”

The Commercial Grade OS Requirement Set requires the following rules.

1. Import with Attributes
   1. validate the security attributes.
   2. if validation succeeds, the OS shall
      1. ensure that the validated security attributes are associated with the imported user data, and
      2. ensure that all applicable access control policies are enforced by the act of importing the user data.
   3. if validation fails, the OS shall associate security attributes with the imported data according to the following authorized administrator specified setting selection:
      1. the DAC and MIC security attributes of the importing subject;
      2. deleting the data.
2. Import without Attributes
   1. associate security attributes with the imported user data according to the following authorized administrator specified setting selection:
      1. the DAC and MIC security attributes of the importing subject;
      2. deleting the data.

This requirement is addressed by the Windows OS as follows.

### Importing with the Data Decryption Field Data Recovery Field security attributes

Recalling from the justification text for addressing the Commercial Grade OS Requirement Set “2.1.3.1” requirement, the [Data Decryption Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) and the [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) are deemed as the security attributes associated with an EFS-protected file. When a subject attempts to gain access to an EFS-protected named file which resides on a remote SMB or WebDAV server, the subject aims to import the clear-text file data belonging to the named file. To validate a [Data Decryption Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) or a [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) as a security attribute, the local EFS key management unit residing within the Windows OS authentication service process (lsass.exe) attempts to use either

* a DPAPI-protected AES 256 master key;
* an EFS certificate public/private key pair;
* an EFS Recovery Agent certificate public/private key pair,

belonging to the user account corresponding to the subject who attempts, to gain access to the EFS-protected named file.

The AES 256 master key protected by the Windows OS DPAPI is attempted to decrypt the AES-encrypted FEK residing in a [Data Decryption Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) as the security attribute validation, which is the above “Importing User Data” rule of “a) 1)”

The EFS certificate RSA private key or the EFS Recovery Agent certificate RSA private key is attempted to decrypt the RSA-encrypted FEK residing in a [Data Decryption Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) or a [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) as the security attribute validation, which is the above “Importing User Data” rule of “a) 1)”.

If the use of the AES 256 master key fails to obtain a valid FEK through the AES-decryption, then the [STATUS\_DECRYPTION\_FAILED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) ([ERROR\_DECRYPTION\_FAILED](http://msdn.microsoft.com/en-us/library/ms819773.aspx)) error code is resulted. If the use of the private key fails to obtain a valid FEK through the RSA-decryption, then the [STATUS\_DECRYPTION\_FAILED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) ([ERROR\_DECRYPTION\_FAILED](http://msdn.microsoft.com/en-us/library/ms819773.aspx)) error code is resulted. This error code signals the security attribute validation failure. In the case where the EFS-protected named file resides in a remote SMB server, the connection to the remote SMB server is closed. In the case where the EFS-protected named file resides in a remote WebDAV server, the local copy of the EFS-protected named file, downloaded from the WebDAV server to the local WebDAV cache directory, is not processed further. In either case, the subject would not be provided a file handle and therefore no clear-text file data is available to the subject. Consequently, the attempted import of the clear-text file data belonging to the named file is aborted. Since no clear-text file data is resulted from the data import abortion, it is deemed as semantically equivalent to the “deleting the data” action stated in the above “Importing User Data” rule of “a) 3) b”.

If the use of the AES 256 master key successfully obtains a valid FEK through the AES-decryption, then the FEK is available to decrypt the EFS-protected named file. If the use of the private key successfully obtains a valid FEK through the RSA-decryption, then the FEK is available to decrypt the EFS-protected named file. As a result, the subject receives the read access to the clear-text file data belonging to the named file, which resides on a remote SMB or WebDAV server. Therefore, the clear-text file data is seen as the user data being imported. Since the access to the clear-text file data is due to the availability of the valid FEK, it is deemed as semantically equivalent to the action of “ensuring that the validated security attributes are associated with the imported user data” stated in the above “Importing User Data” rule of “a) 2) a”.

To complete the import of the clear-text file data as persistent user data on the local Windows OS machine, the subject writes the clear-text file data into another target named file residing in the local Windows OS machine through a means such as “copy and paste” or “save as” of an application of the subject. In order to write into the target named file, the subject needs to obtain a valid file handle, which is granted with at least the write access, to the target named file. The authorization to grant the write access file handle to the subject has to follow the Mandatory Integrity Control Policy and the Discretionary Access Control Policy applicable to NTFS file directory objects and NTFS file objects mentioned in the justification text for addressing the Commercial Grade OS Requirement Set “2.2.1.1” and “2.1.1.1” requirements respectively.

Specifically, if the target named file already exists, the authorization follows

* the Mandatory Integrity Control Policy “[SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)” rule, stated in the justification text for addressing the Commercial Grade OS Requirement Set “2.2.1.5” requirement, for NTFS file objects;
* the Discretionary Access Control Policy rules, stated in the justification text for addressing the Commercial Grade OS Requirement Set “2.1.1.6” requirement, for NTFS file objects.

If the target named file does not exist, the target named file to contain the clear-text file data being imported is new. It needs to reside in the parent NTFS directory object specified in the full path name of the target named file as a new child NTFS file object. The authorization follows

* the Mandatory Integrity Control Policy “[SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)” rule, stated in the justification text for addressing the Commercial Grade OS Requirement Set “2.2.1.5” requirement, for file directory objects;
* the Discretionary Access Control Policy rules, stated in the justification text for addressing the Commercial Grade OS Requirement Set “2.1.1.6” requirement, for NTFS file directory objects.

Therefore, the authorization is deemed as semantically equivalent to the action of “ensuring that all applicable access control policies are enforced by the act of importing the user data” stated in the above “Importing User Data” rule of “a) 2) b”.

In the case where

* the target named file is new;
* the subject does not provide a [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE for the file;
* the parent NTFS directory object containing the target named file does not have an inheritable [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE for its child file objects,

then the MIC enforcement algorithm uses the following default [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE of:

* an integrity level value of MandatoryLevelMedium (2) (SepDefaultMandatorySid);
* the write ([SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class.

Effectively, it implies that the target named file is assigned the above default [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE.

In the case where

* the target named file is new;
* the subject does not provide a [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE for the file;
* the parent NTFS directory object containing the target named file has an inheritable [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE for its child file objects,

then the target named file inherits the inheritable [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE from its parent NTFS directory object.

In the case where

* the target named file is new;
* the subject does not provide a DACL for the file;
* the parent NTFS directory object containing the target named file does not have an inheritable DAC policy enforcement ACE for its child file objects,

then the DACL for the target named file is the subject’s default DACL.

In the case where

* the target named file is new;
* the subject does not provide a [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE for the file,
* the parent NTFS directory object containing the target named file has an inheritable DAC policy enforcement ACE for its child file objects,

then the target named file inherits the inheritable DAC policy enforcement ACE from its parent NTFS directory object.

### Importing without any security attribute

When a subject attempts to gain access to a non-EFS-protected source named file which resides on a remote SMB or WebDAV server, the subject aims to import simply the file data belonging to the source named file. The source named file does not have an “$EFS” metadata stream. There is no Data Decryption Field or Data Recovery Field security attribute associated with the source named file. To complete the import of the file data as persistent user data on the local Windows OS machine, the subject writes the file data into another target named file residing in the local Windows OS machine through a means such as “copy and paste” or “save as” of an application of the subject. In order to write into the target named file, the subject needs to obtain a valid file handle, which is granted with at least the write access, to the target named file.

In the case where

* the target named file is new;
* the subject does not provide a [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE for the file;
* the parent NTFS directory object containing the target named file does not have an inheritable [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE for its child file objects,

then the MIC enforcement algorithm uses the following default [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE of:

* an integrity level value of MandatoryLevelMedium (2) (SepDefaultMandatorySid);
* the write ([SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx)) access operation class.

Effectively, it implies that the target named file is assigned the above default [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE.

In the case where

* the target named file is new;
* the subject does not provide a [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE for the file,
* the parent NTFS directory object containing the target named file has an inheritable [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE for its child file objects,

then the target named file inherits the inheritable [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE from its parent NTFS directory object.

In the case where

* the target named file is new;
* the subject does not provide a DACL for the file;
* the parent NTFS directory object containing the target named file does not have an inheritable DAC policy enforcement ACE for its child file objects,

then the DACL for the target named file is the subject’s default DACL.

In the case where

* the target named file is new;
* the subject does not provide a [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE for the file;
* the parent NTFS directory object containing the target named file has an inheritable DAC policy enforcement ACE for its child file objects,

then the target named file inherits the inheritable DAC policy enforcement ACE from its parent NTFS directory object.

The target named file contains the file data belonging to the source named file as the imported user data.

As a result, in the case where the target named file is new,

* the default [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE;
* the subject’s default DACL;
* the inheritable [SYSTEM\_MANDATORY\_LABEL\_ACE](http://msdn.microsoft.com/en-us/library/aa965848(VS.85).aspx) ACE from the target named file’s parent NTFS directory object;
* the inheritable DAC policy enforcement ACEs from the target named file’s parent NTFS directory object

are security attributes that could be associated with the target named file. This security attribute association is deemed as a semantic alternative to the “DAC and MIC security attributes of the importing subject” action stated in the above “Importing User Data” rule of “b) 1)”.

Finally, in the case where the target named file already exists, the target named file has its MIC and DAC security attributes already. Therefore, the “DAC and MIC security attributes of the importing subject” action stated in the above “Importing User Data” rule of “b) 1)” is not necessary.

Given the above explanations, this Commercial Grade OS Requirement Set “6.2.1.1” requirement is met.

# Meeting the “Import/Export of Data Imported Data Management Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual management requirement under the heading of “Import/Export of Data Imported Data Management Requirements”. It is listed as “6.2.2.1”.

## Addressing 6.2.2.1 “The OS shall provide authorized administrators with the ability to select the technique for handling imported data with attributes that cannot be validated”

This requirement is addressed by the Windows OS as follows.

As described in the justification text for addressing the Commercial Grade OS Requirement Set “6.2.1.1” requirement, the following behaviors are fixed.

* The attempted import of the clear-text file data belonging to an EFS-protected named file is aborted in the case where the [Data Decryption Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) and [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) security attributes associated with the named file cannot be validated.
* In the case where a target named file contains the file data belonging to the non-EFS-protected source named file as the imported user data, specific MIC and DAC security attributes are associated with the target named file.

Therefore, there is no need for an administrator to alter the above behaviors. As a result, this Commercial Grade OS Requirement Set “6.2.2.1” requirement is moot for the Windows OS.

# Meeting the “Import/Export of Data Imported Data Audit Requirements”

In the Commercial Grade OS Requirement Set, there are 2 individual audit requirements under the heading of “Import/Export of Data Imported Data Audit Requirements”. They are listed as “6.2.3.n”, where n = 1 and 2.

## Addressing 6.2.3.1 “The OS shall provide the ability to audit the failure of security attribute validation”

There is no generation of an audit record to record a failure of the validation of a [Data Decryption Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) or a [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) security attribute associated with an EFS-protected named file. This auditing is deemed as unnecessary as the outcome is not sufficiently different from any other kind of user data unavailability situations from the end-user experience perspective.

The attempted import of the clear-text file data belonging to an EFS-protected named file is simply aborted in the case where the [Data Decryption Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) and [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) security attributes associated with the named file cannot be validated.

## Addressing 6.2.3.2 “The OS shall provide the ability to audit the administrator selected configuration for handling imported data with attributes that cannot be validated”

This requirement is addressed by the Windows OS as follows.

As described in the justification text for addressing the Commercial Grade OS Requirement Set “6.2.1.1” requirement, the following behaviors are fixed.

* The attempted import of the clear-text file data belonging to an EFS-protected named file is aborted in the case where the [Data Decryption Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) and [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) security attributes associated with the named file cannot be validated.
* In the case where a target named file contains the file data belonging to the non-EFS-protected source named file as the imported user data, specific MIC and DAC security attributes are associated with the target named file.

Therefore, there is no need for an administrator to alter the above behaviors. Since the alteration of the above behaviors is not possible, the auditing of this alteration is deemed as unnecessary. Consequently, this Commercial Grade OS Requirement Set “6.2.3.2” requirement is moot for the Windows OS.

# Meeting the “Import/Export of Data Trusted Channels Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 3 individual functional requirements under the heading of “Import/Export of Data Trusted Channels Functional Requirements”. They are listed as “6.3.1.n”, where n = 1, 2 and 3.

## Addressing 6.3.1.1 “The OS shall provide a communication path between itself and users that is logically distinct from other communication paths and provides assured identification of the OS to the requesting user”

This requirement is addressed by the Windows OS as follows.

The “CTRL + ALT + DELETE” hot key is the means that the Windows OS uses to implement the trusted path between itself and (local interactive) users. Every Windows OS window manager (win32k.sys) instance is responsible for the hot key registration and the corresponding maintenance within its window terminal session. In every window terminal session, only the first Windows OS windowstation (namely “Winsta0”) receives IO via a keyboard, mouse, human interface device (HID) and video monitors. A window logon state maintaining service (winlogon.exe) instance requests the creation of the “Winsta0” windowstation during its initialization. As the “Winsta0” windowstation is created, the Windows OS window manager instance initializes the input system by executing the “Raw Input” Windows OS thread in the security context of the local system. The “Raw Input” thread processes raw keyboard, mouse and HID input and otherwise waits for the shutdown and other events in a listening loop. Before the start of the raw input processing, the “Raw Input” thread registers the following hot keys with a callback function:

* [VK\_DELETE](http://msdn.microsoft.com/en-us/library/ms645540.aspx) with the modifiers: MOD\_SAS | [MOD\_CONTROL](http://msdn.microsoft.com/en-us/library/ms646309.aspx) | [MOD\_ALT](http://msdn.microsoft.com/en-us/library/ms646309.aspx)
  + this is the “CTRL + ALT + DELETE” “trusted path” hot key in the Windows OS;
* [VK\_ESCAPE](http://msdn.microsoft.com/en-us/library/ms645540.aspx) with the modifiers: [MOD\_CONTROL](http://msdn.microsoft.com/en-us/library/ms646309.aspx) | [MOD\_SHIFT](http://msdn.microsoft.com/en-us/library/ms646309.aspx)
  + this is the default hot key for bringing up the task manager (taskmgr.exe);
* "L" with the modifiers: MOD\_SAS | [MOD\_WIN](http://msdn.microsoft.com/en-us/library/ms646309.aspx)
  + this is the default hot key for initiating the locking of the display areas;
* "U" with the modifiers: [MOD\_WIN](http://msdn.microsoft.com/en-us/library/ms646309.aspx)
  + this is the default hot key for bringing up the “Ease of Access Center” control panel applet.

The hotkey registration means that whenever one of the above registered hot keys arrives through the raw keyboard, mouse and HID input processing, the corresponding registered hot key callback function is activated. The hot key callback function does nothing but notifies the window logon state maintaining service instance, residing in the same window terminal session, about the registered hot key arrival.

The registered “CTRL + ALT + DELETE” hot key trusted path is logically distinct from other communication paths and provides assured identification of the Windows OS to the requesting (local interactive) user because of the following behaviors of every Windows OS window manager instance and its accompanied window logon state maintaining service instance.

* While its window terminal session is connected to the local display terminal for displaying the contents (i.e. window graphical interfaces) of its display areas and accepting user input, the Windows OS window manager instance mediates input data generated by local hardware devices such as keyboards, mice, and other Human Interface Devices (HID), through the “Raw Input” Windows OS thread, before delivering them to its consumer subjects as authentic (local interactive) user input.
* In a disconnected window terminal session, its Windows OS window manager instance does not receive input data generated by local hardware devices such as keyboards, mice, and other Human Interface Devices (HID), as explained in the following section of this paper.
  + “Human interface device input and video output separation maintained by the Windows OS window manager(s)”.
* Mouse events and keyboard events arriving through the local mouse and keyboard devices are not marked as injected.
* Raw HID events are delivered through [WM\_INPUT](http://msdn.microsoft.com/en-us/library/ms645590(VS.85).aspx) messages to the windows whose process owner subjects have successfully been registered for receiving raw HID events.
* Mouse events are not deemed as any registered hot key (trusted path) input events.
* If a subject requests to register a hot key for the specified window object so that the window object would receive hot key window messages generated by the hot key being registered, then
  + the requesting subject must have the write attributes access to all the display areas;
  + the requesting subject must have the read object and write object access to the display area where the window object exists in;
  + the integrity level (IL) of the requesting subject must dominate the IL of the owner of the specified window object;
  + the hot key being registered must not be the already registered hot keys (including the “CTRL + ALT + DELETE” “trusted path” hot key).
* If a subject requests to register a system wide hot key so that hot key window messages generated by the hot key being registered would be posted to the message queue that the subject is associated with, then
  + the requesting subject must have the write attributes access to all the display areas;
  + the hot key being registered must not be the already registered hot keys (including the “CTRL + ALT + DELETE” “trusted path” hot key).
* If a subject requests to unregister a hot key previously registered using the specified window object, then
  + the requesting subject must be the same subject who has registered the hot key;
  + the requesting subject must have the read object and write object access to the display area where the window object exists in;
  + the integrity level (IL) of the requesting subject must dominate the IL of the owner of the specified window object.
* If a subject requests to unregister a hot key previously registered, then
  + the requesting subject must be the same subject who has registered the hot key.
* A subject, other than the local system, is not allowed to request the Windows OS window manager instance to reserve specific keyboard keys for the specified console window so that the keys are exempted for hot key behaviors when they arrive at the window.
* A successfully installed and allowed global low level keyboard or mouse hook procedure does not block a local interactive user from pushing the “CTRL + ALT + DELETE” “trusted path” hot key.
* If a subject requests to insert injected keyboard events serially into the input stream for being delivered to window objects, then
  + the requesting subject must have either
    - the read object and write object access to the current display area;
    - the journal playback access to the current display area;
  + either the IL of the requesting subject must dominate the IL of the foreground message queue or the consumer subject has been granted the UI access right;
  + the injected keyboard events must not be registered hot key (trusted path) input events;
  + another subject must not have blocked keyboard and mouse input events from being delivered to window objects;
  + a successfully installed and allowed low level keyboard hook procedure must not block the injected keyboard events.
* A local interactive user pushing the “CTRL + ALT + DELETE” “trusted path” hot key always unblock keyboard and mouse input events from not being delivered to window objects.
* If the “Welcome” state transitions to the “Display legal notices” state of the window logon state maintaining service instance, then there must be either:
  + the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
  + the notification of a trusted path simulation from an authorized subject.
* If the “User logged on” state transitions to the “Trusted path activated” state of the window logon state maintaining service instance, then there must be either:
  + the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
  + the notification of a trusted path simulation from an authorized subject.
* If the “Locked” state transitions to the “Request to unlock” state of the window logon state maintaining service instance, then either:
  + the instance of the window logon state maintaining service must operate in the remote window terminal session case;
  + the “[disable trusted path](http://technet2.microsoft.com/windowsserver/en/library/83e7e44d-84fe-4e8b-9e5c-df97609eb8601033.mspx?mfr=true)” policy must have been enabled;
  + there must have been either:
    - the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
    - the notification of a trusted path simulation from an authorized subject.
* If the “Inactivity timeout handler during locked” state transitions to the “Locked” state and then to the “Request to unlock” state of the window logon state maintaining service instance, then either:
  + the instance of the window logon state maintaining service must operate in the remote window terminal session case;
  + the “[disable trusted path](http://technet2.microsoft.com/windowsserver/en/library/83e7e44d-84fe-4e8b-9e5c-df97609eb8601033.mspx?mfr=true)” policy must have been enabled;
  + there must have been either:
    - the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
    - the notification of a trusted path simulation from an authorized subject.
* If the “Inactivity timeout to run screensaver” state transitions to the “Welcome” state of the window logon state maintaining service instance, then either:
  + the administrator specified screen saver application process started by the instance of the window logon state maintaining service must have terminated;
  + the Local Window Terminal Service must have notified disconnection of the window terminal session where the instance of the window logon state maintaining service resides in;
  + the local shutdown initiation service (as part of ininit.exe) must have notified a logoff or shutdown;
  + a logoff or shutdown must have been triggered internally within the instance of the window logon state maintaining service;
  + there must have been either:
    - the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
    - the notification of a trusted path simulation from an authorized subject;
  + there must have been a notification originated from the Windows OS power manager or another authorized source to request the window logon user interface service (aka LogonUI.exe) to display the below messages:
    - “Windows is shutting down...” corresponding to data saving;
    - “Undocking and going to sleep...”;
    - “Hibernating...”
    - “Going to sleep...” corresponding to the entering of the “standing by” Windows OS system power state.
* If the “Post logged on inactivity timeout handler” state transitions to the “Trusted path activated” state of the window logon state maintaining service instance, then
  + there must have been either:
    - the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
    - the notification of a trusted path simulation from an authorized subject.
* If a subject requests the window logon state maintaining service instance to make a trusted path simulation, then
  + the requesting subject must reside locally on the same physical Windows OS machine;
  + the requesting subject must reside in the same window terminal session as the window logon state maintaining service instance;
  + the “software trusted path generation” policy must have been set to the “Ease of Access applications” value so that “Ease of Access” applications, running in the current logged on user account’s security context with the UI Access right in the same session as the window logon state maintaining service instance, can request for a trusted path simulation;
  + the requesting subject must be the currently interactive logged on user;
  + the requesting subject must have the UI Access right.
* If a subject requests to set a value for the “software trusted path generation” policy (aka the “Disable or enable software Secure Attention Sequence” policy), as a per user behavior item for the user account that the subject represents, then
  + the behavior item must be allowed by the administrator specified policy.
* The following values are supported by the “software trusted path generation” policy:
  + If this policy is set to "None" (i.e. the value = 0), then user mode software cannot request for a trusted path simulation, in the case where the window logon state maintaining service instance is residing a local window terminal session.
  + If this policy is set to “Services” (i.e. the value = 1), then local system services running in window terminal session 0 can request for a trusted path simulation, in the case where the window logon state maintaining service instance is residing a local window terminal session.
  + If this policy is set to "Ease of Access applications" (i.e. the value = 2), then “Ease of Access” applications running in the current logged on user account’s security context with the UI Access right in the same window terminal session as the window logon state maintaining service instance can request for a trusted path simulation.
* By default, the “software trusted path generation” policy is set to “None” (i.e. the value = 0).
* A non-kernel mode subject, other than the local system, is not permitted to send either of the following messages to the window logon state maintaining service instance, in the case where the window logon state maintaining service instance is residing a local window terminal session.
  + The notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session.
  + The notification of a trusted path simulation from an authorized subject.
* The (remote) interactive user logging onto a remote window terminal session can request the Windows OS remote window terminal service (termsrv.dll) to send the notification of a trusted path simulation to the window logon state maintaining service instance of the window terminal session, causing the window logon user interface service of the window terminal session to display the following options in a dialog on the secure display area of the window terminal session for the interactive user to select:
  + Log off from the window terminal session that is currently visible remotely;
  + Launch the accessibility tool;
  + Lock the display areas of the (remote) window terminal session from visibility;
  + Change the logon user’ password;
  + Start the task manager (taskmgr.exe).

Given the above explanations, this Commercial Grade OS Requirement Set “6.3.1.1” requirement is met.

## Addressing 6.3.1.2 “The OS shall permit users to initiate communication via a trusted path”

This requirement is addressed by the Windows OS as follows.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “6.3.1.1” requirement, the registered “CTRL + ALT + DELETE” “trusted path” hot key is available to local interactive users to push from the locally attached keyboard in a window terminal session that is connected to the local display terminal for displaying the contents (i.e. window graphical interfaces) of its display areas and accepting user input.

Given the above explanations, this Commercial Grade OS Requirement Set “6.3.1.2” requirement is met.

## Addressing 6.3.1.3 “The OS shall require the use of a trusted path for all user operations involving authentication data”

This requirement is addressed by the Windows OS as follows.

In the justification text for addressing the Commercial Grade OS Requirement Set “6.3.1.1” requirement, the state transitions of every window logon state maintaining service instance that could be caused by the following two notifications have been described.

* the notification of an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) which has been mediated by the local Windows OS window manager of the same window terminal session;
* the notification of a trusted path simulation from an authorized subject.

The following three states of the window logon state maintaining service instance are the target states of these state transitions caused by an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) or a trusted path simulation from an authorized subject.

* the “Display legal notices” state;
* the “Trusted path activated” state;
* the “Request to unlock” state.

Recall from the “Transitions from the “Display legal notices” state” section,

* if the “Display legal notices” state transitions to the “Request logon credentials” state of the window logon state maintaining service instance, then either:
  + there must have been the interactive user’s acknowledgement as his response to the displayed “legal notices” message box message on the secure display area;
  + the administrator must have not specified any “legal notices”.

From all the state transitions of a window logon state maintaining service instance documented in the [Microsoft publication: “Security Functional Assertions of the “User Interaction based on Windowing” Scenario of a Modern Operating System”](http://download.microsoft.com/download/e/d/b/edbb17fb-580d-49a4-b66c-8726cf446a86/User%20Interaction.doc), it can be seen that the following states can be reached only due to either an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) or a trusted path simulation from an authorized subject, in the case where the window logon state maintaining service instance resides in the window terminal session that is connected to the local display terminal for displaying the contents (i.e. window graphical interfaces) of its display areas and accepting user input:

* the “Request logon credentials” state;
* the “Trusted path activated” state;
* the “Request to unlock” state.

In the case where the window logon state maintaining service instance resides in the window terminal session that is connected to the local display terminal for displaying the contents (i.e. window graphical interfaces) of its display areas and accepting user input, these three states are the only states which would lead to an interactive dialog appearing on the secure display area to request the local interactive user to supply the user’s credential information (such as the user name with a password or the smart card PIN) corresponding to a credential provider installed on the local machine.

As a result, the above has shown that the Windows OS requires either an arrival of the registered trusted path hot key (e.g. CTRL + ALT + DELETE) or a trusted path simulation from an authorized subject for an interactive dialog to appear on the secure display area for requesting the local interactive user to supply the user’s credential information (such as the user name with a password or the smart card PIN) corresponding to a credential provider installed on the local machine.

Given the above explanations, this Commercial Grade OS Requirement Set “6.3.1.3” requirement is met.

# Meeting the “Import/Export of Data Trusted Channels Management Requirements”

In the Commercial Grade OS Requirement Set, there is no individual management requirement under the heading of “Import/Export of Data Trusted Channels Management Requirements”.

# Meeting the “Import/Export of Data Trusted Channels Audit Requirements”

In the Commercial Grade OS Requirement Set, there is no individual audit requirement under the heading of “Import/Export of Data Trusted Channels Audit Requirements”.

# Meeting the “Revocation Access Revocation Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 3 individual functional requirements under the heading of “Revocation Access Revocation Functional Requirements”. They are listed as “7.1.1.n”, where n = 1, 2 and 3.

## Addressing 7.1.1.1 “The OS shall enforce the revocation of security relevant attributes associated with named objects when access checks are made”

This requirement is addressed by the Windows OS as follows.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “2.1.2.1” requirement, the [WRITE\_DAC](http://msdn.microsoft.com/en-us/library/aa379607.aspx) is one of the standard access control elements applicable to every named object. It corresponds to the right to modify the DACL of a named object.

Similarly, as explained in the justification text for addressing the Commercial Grade OS Requirement Set “2.2.2.2” requirement, a security descriptor, including a MACL, may be modified by an authorized subject. The authorized subject must possess the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) to request the responsible resource manager to update the MIC security attributes of a specific named object.

Therefore, to revoke elements of the DACL associated with a named object, the authorized subject must possess the WRITE\_DAC in the DACL. Similarly, to revoke elements of the MACL associated with a named object, the authorized subject must possess the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK).

Whenever a resource manager makes an access check for a named object, it typically acquires a fresh copy of the security descriptor associated with the named object[[29]](#footnote-30).

Assume that the successful revocation of DACL or MACL elements has been made to the security descriptor before the resource manager acquires a fresh copy of the security descriptor for the access check. The resource manager actually enforces the revocation of DACL or MACL elements associated with the named object when the access check for the named object is made.

As a result, this Commercial Grade OS Requirement Set “7.1.1.1” requirement is met.

## Addressing 7.1.1.2 “The OS shall enforce the revocation of security relevant attributes associated with users at user session establishment”

This requirement is addressed by the Windows OS as follows.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement, the following security attributes associated with a user account are used by the Windows OS authentication service in the authentication decision for a user attempting to log onto the user account:

* [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It specifies the [hours](http://technet2.microsoft.com/windowsserver/en/library/a0f007ba-b3ef-4c1f-8836-9087fe5eb08d1033.mspx?mfr=true) that the user account is allowed to be logged on in the Windows OS domain;
  + By default, a standard user does not possess the right to update this attribute value;
  + When the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) security attribute is changed successfully by an administrator, the new value is recorded in the “Logon Hours” informational item of the [Event ID 4738](Event%20ID%204767%20(SE_AUDITID_ACCOUNT_UNLOCKED/SE_AUDITID_ETW_ACCOUNT_UNLOCKED)) security audit record “A user account was changed” (SE\_AUDITID\_USER\_CHANGE/ SE\_AUDITID\_ETW\_USER\_CHANGE) as generated by the Windows OS SAM;
  + If the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) security attribute has been updated in the revocation so that the current time is not in the scope of the logon hours, the window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the user account is not allowed to logon at the current time due to the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) based time restriction policy assigned to the account by an administrator ([STATUS\_INVALID\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/cc704588.aspx)):
    - “Your account has time restrictions that prevent you from logging on at this time.”
* Passwords security attributes
  + [ATT\_UNICODE\_PWD](http://msdn.microsoft.com/en-us/library/ms680513(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
    - It stores the NTLM password hash in the format suitable for the user authentication being conducted by the NTLM security provider;
    - If the user account is domain based, it can be updated by an administrator without the knowledge of the current password as there is the ACCESS\_ALLOWED\_OBJECT\_ACE ACE for the GUID\_CONTROL\_UserForceChangePassword property set to allow the administrator in the user account object’s default security descriptor;
    - If the user account is locally defined, it can be updated by an administrator without the knowledge of the current password as there is the ACCESS\_ALLOWED\_ACE ACE for USER\_FORCE\_PASSWORD\_CHANGE to allow the administrator in the user account object’s default security descriptor;
  + [ATT\_SUPPLEMENTAL\_CREDENTIALS](http://msdn.microsoft.com/en-us/library/ms679920(VS.85).aspx) (if the user account is domain based)
    - It [stores](http://msdn.microsoft.com/en-us/library/cc211940.aspx) the password-derived keys (e.g. KdcBuildKerbCredentialsFromPassword()), the password hashes, or the clear text password in a format suitable for the user authentication being conducted by the Kerberos KDC and Kerberos security provider and the Digest security provider;
  + If the [ATT\_UNICODE\_PWD](http://msdn.microsoft.com/en-us/library/ms680513(VS.85).aspx) and the [ATT\_SUPPLEMENTAL\_CREDENTIALS](http://msdn.microsoft.com/en-us/library/ms679920(VS.85).aspx) security attributes have been changed in the revocation, the window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports a logon failure status ([STATUS\_LOGON\_FAILURE](http://msdn.microsoft.com/en-us/library/cc704588.aspx)) and a wrong password status ([STATUS\_WRONG\_PASSWORD](http://msdn.microsoft.com/en-us/library/cc704588.aspx)):
    - “The user name or password is incorrect.”
  + When the [ATT\_UNICODE\_PWD](http://msdn.microsoft.com/en-us/library/ms680513(VS.85).aspx) and the [ATT\_SUPPLEMENTAL\_CREDENTIALS](http://msdn.microsoft.com/en-us/library/ms679920(VS.85).aspx) attributes are updated successfully by an administrator, the [Event ID 4724](http://support.microsoft.com/kb/947226) security audit record “An attempt was made to reset an account's password” (SE\_AUDITID\_USER\_PWD\_SET/SE\_AUDITID\_ETW\_USER\_PWD\_SET) is generated by the Windows OS SAM;
* [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) (if the user account is domain based)
  + It contains associated mappings for X.509 public key certificates or external Kerberos user accounts to the user account for the purpose of public key certificate or smart card based user authentication;
  + By default, a standard user does not possess the right to update this attribute value;
  + When the [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) attribute is changed successfully by an administrator, the change is recorded in the “Directory Object Attribute Value” informational item of the [Event ID 5136](http://support.microsoft.com/kb/947226) security audit record “A directory service object was modified” (SE\_AUDITID\_ETW\_DS\_OBJECT\_MODIFY\_value) as generated by the Windows OS Active Directory;
  + If the [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) security attribute has been revoked in the revocation, the window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports a logon failure status ([STATUS\_LOGON\_FAILURE](http://msdn.microsoft.com/en-us/library/cc704588.aspx)) and a “non existence of a specific user account” status ([STATUS\_NO\_SUCH\_USER](http://msdn.microsoft.com/en-us/library/cc704588.aspx)):
    - “The user name or password is incorrect.”
* [ATT\_USER\_ACCOUNT\_CONTROL](http://msdn.microsoft.com/en-us/library/ms680832(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + Among other code values, it contains specifically the USER\_ACCOUNT\_DISABLED code value for indicating that the specific user account is disabled;
  + By default, a standard user does not possess the right to update this attribute value;
  + When the USER\_ACCOUNT\_DISABLED code value in the [ATT\_USER\_ACCOUNT\_CONTROL](http://msdn.microsoft.com/en-us/library/ms680832(VS.85).aspx) attribute is changed successfully by an administrator, the change is recorded in the “Old UAC Value”, “New UAC Value” and “User Account Control” informational items of the [Event ID 4738](Event%20ID%204767%20(SE_AUDITID_ACCOUNT_UNLOCKED/SE_AUDITID_ETW_ACCOUNT_UNLOCKED)) security audit record “A user account was changed” (SE\_AUDITID\_USER\_CHANGE/ SE\_AUDITID\_ETW\_USER\_CHANGE) as generated by the Windows OS SAM;
  + If the USER\_ACCOUNT\_DISABLED code value of the [ATT\_USER\_ACCOUNT\_CONTROL](http://msdn.microsoft.com/en-us/library/ms680832(VS.85).aspx) security attribute has been set in the revocation, the window logon user interface service displays the following message on the secure display area (during an initial interactive logon attempt or during a display area unlock attempt) when the Windows OS Authentication Service reports the status that the user account is disabled ([STATUS\_ACCOUNT\_DISABLED](http://msdn.microsoft.com/en-us/library/cc704588.aspx)):
    - “Your account has been disabled.”

Therefore, the revocation of one of the aforementioned security attributes is enforced during the user session establishment (i.e. during an initial interactive logon attempt).

The Windows OS authentication service decides the authentication of a logging on user to the specified user account during the user’s session establishment (i.e. during an initial interactive logon attempt). As explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.8” requirement, the Windows OS security reference monitor (SRM) creates an access token for the user, upon the request from the Windows OS authentication service. The following elements, among the others, are included in the access token:

* the list of groups (in the form of group object SIDs) that the logged on user account is a member of directly or indirectly;
* the list of Windows OS privileges assigned to the logged on user account.

These elements are the security attributes associated with the user account. They may be revoked by an administrator. Only the fresh values of these elements are used for their inclusion in the user access token. The value of an element is fresh after an administrator successfully revokes the element. The inclusion of these elements in the access token for the user is the actual enforcement of the revocation of group membership and Windows OS privileges for the logging on user.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement, a group membership is defined by the following security attribute:

* [ATT\_IS\_MEMBER\_OF\_DL](http://msdn.microsoft.com/en-us/library/ms677099(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It contains the distinguished names of the groups to which this user account directly belongs to;
  + Group nesting is possible in Windows OS;
  + A [group object](http://msdn.microsoft.com/en-us/library/ms682251(VS.85).aspx) also has its own ATT\_IS\_MEMBER\_OF\_DL attribute and so SAM is responsible to conduct the group membership expansion;
  + By default, a standard user does not possess the right to update this attribute value;
  + When the [ATT\_IS\_MEMBER\_OF\_DL](http://msdn.microsoft.com/en-us/library/ms677099(VS.85).aspx) attribute is updated successfully due to a group membership removal revocation action of an administrator, one of the following Event ID security audit records is generated by the Windows OS SAM:
    - [Event ID 4733](http://support.microsoft.com/kb/947226) security audit record “A member was removed from a security-enabled local group” (SE\_AUDITID\_LOCAL\_GROUP\_REM/SE\_AUDITID\_ETW\_LOCAL\_GROUP\_REM), indicating
      * Name of the member removed;
      * ID of the member removed;
    - [Event ID 4729](http://support.microsoft.com/kb/947226) security audit record “A member was removed from a security-enabled global group” (SE\_AUDITID\_GLOBAL\_GROUP\_REM/SE\_AUDITID\_ETW\_GLOBAL\_GROUP\_REM), indicating
      * Name of the member removed;
      * ID of the member removed;
    - [Event ID 4757](http://support.microsoft.com/kb/947226) security audit record “A member was removed from a security-enabled universal group” (SE\_AUDITID\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_REM/ SE\_AUDITID\_ETW\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_REM), indicating
      * Name of the member removed;
      * ID of the member removed.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.8” requirement, the Windows OS authentication service determines the Windows OS privilege set and the system access for the logging on user account due to the user SID and the expanded final group membership of the user account from its policy database (which is maintained within its own Windows OS process). The list of Windows OS privileges and the list of Windows OS system access are described in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.8” requirement.

Finally, as described in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.3.2” requirement, the policy database component of the Windows OS authentication service is responsible to generate the following security audit records when the Windows OS privilege (i.e. user right) and system access assignment for a user account or a group is revoked by an administrator.

* [Event ID 4705](http://support.microsoft.com/kb/947226) security audit record “A user right was removed” (SE\_AUDITID\_ETW\_USER\_RIGHT\_REMOVED\_value)
  + Specific rights (i.e. Windows OS privileges) removed;
* [Event ID 4718](http://support.microsoft.com/kb/947226) security audit record “System security access was removed from an account” (SE\_AUDITID\_ETW\_SYSTEM\_ACCESS\_REMOVED\_value)
  + Specific security access removed.

The above have completed the justification for addressing this Commercial Grade OS Requirement Set “7.1.1.2” requirement.

## Addressing 7.1.1.3 “The OS shall immediately terminate all subjects associated with deleted user accounts”

This requirement is addressed by the Windows OS as follows.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement, the Windows OS defines user account objects in the Active Directory (for Windows OS domain wide user accounts) and in the Windows OS SAM of an individual machine (for Windows OS local user accounts on the machine).

To delete a specified user account object, which is defined locally in the SAM, the SAM [SamrDeleteUser()](http://msdn.microsoft.com/en-us/library/cc212070.aspx) interface is used by the user account object deletion requester. However, the requester must possess the [DELETE](http://msdn.microsoft.com/en-us/library/aa379607.aspx) standard access right. By default, this DELETE standard access right is granted to

* an administrator;
* an account operator.

Unfortunately, the user account deletion operation of the SAM [SamrDeleteUser()](http://msdn.microsoft.com/en-us/library/cc212070.aspx) interface does not incorporate the termination of running Windows OS processes (i.e. subjects) associated with the deleted user account before its completion. However, the administrator, who requested the deletion operation, is evidently active in the Windows OS machine where the deleted user account resided. If the user account has not been logged onto before the user account deletion, then there would be no running Windows OS processes associated with the user account. If the user account is logged onto when the user account deletion occurs, then the administrator manually can use the [ExitWindowsEx() API](http://msdn.microsoft.com/en-us/library/aa376868(VS.85).aspx) with the EWX\_FORCEIFHUNG flag to initiate a forced logging off operation for the user account, and therefore attempt to terminate all running Windows OS processes associated with the user account within the Windows OS machine.

To delete a specified user account object, which is defined in the Active Directory as a child object in the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx), the Windows OS LDAP server provides the LDAP interface for a subject to attempt the deletion operation using the [LDAP\_DELETE\_CMD](http://msdn.microsoft.com/en-us/library/aa366554(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_DELETE\_TREE right in the security descriptor for the user account object. By default, the security descriptor for a user account object grants the RIGHT\_DS\_DELETE\_TREE right to

* an administrator;
* local system;
* an account operator.

Unfortunately, the user account deletion operation of the LDAP interface using the [LDAP\_DELETE\_CMD](http://msdn.microsoft.com/en-us/library/aa366554(VS.85).aspx) command does not incorporate the termination of running Windows OS processes (i.e. subjects) associated with the deleted user account before its completion. In fact, the user account deletion occurs in the Windows OS Active Directory (i.e. the cooperating domain controllers). On the other hand, the user account is logged onto in one or more logging on sessions within the distributed environment realized by the Windows OS domain/forest. A logging on session occurs in a managed machine of the Windows OS domain/forest.

If there is not a logging on session, where the user account is logged onto, within the Windows OS domain/forest, before the user account deletion, then there would be no running Windows OS processes associated with the user account.

Assume that there is a logging on session, where the user account is logged onto, within the Windows OS domain/forest, when the user account deletion occurs in the Windows OS Active Directory. There is no guarantee that the Windows OS machine, where the logging on session exists, is connected.

If the Windows OS machine is not connected, then a remote initiation of a forced logging off operation for the user account on the machine is not possible. The administrator needs to obtain the physical control of the machine in order to terminate the running Windows OS processes associated with the deleted user account within the machine. For example, after obtaining the physical control, the administrator can reboot the machine.

If the machine is connected, then a remote initiation of a forced logging off operation for the user account on the machine is possible. The administrator manually can use either

* the [ExitWindowsEx() API](http://msdn.microsoft.com/en-us/library/aa376868(VS.85).aspx) with the EWX\_FORCEIFHUNG flag;
* the [WTSLogoffSession() API](http://msdn.microsoft.com/en-us/library/aa383836.aspx) specifying the window terminal session of the logging on session,

to initiate a forced logging off operation for the user account, and therefore attempt to terminate all running Windows OS processes associated with the user account within the connected machine.

In summary, the Windows OS does not terminate all running Windows OS processes (i.e. subjects) associated with the deleted user account as the user account is deleted. However, as long as the machines, where there is a logging on session for the deleted user account, are connected, the Windows OS provides authorized administrators the ability to initiate a forced logging off operation for the user account on the machines. However, if a machine, where there is a logging on session for the deleted user account, is not connected, then the administrator needs to obtain the physical control of the machine in order to terminate the running Windows OS processes associated with the deleted user account within the machine.

As a result, we believe that the intents behind this Commercial Grade OS Requirement Set “7.1.1.3” requirement are sufficiently addressed.

# Meeting the “Revocation Access Revocation Management Requirements”

In the Commercial Grade OS Requirement Set, there are 4 individual management requirements under the heading of “Revocation Access Revocation Management Requirements”. They are listed as “7.1.2.n”, where n = 1, 2, 3 and 4.

## Addressing 7.1.2.1 “The OS shall allow authorized administrators, object owners, and users with the DAC change attribute the ability to revoke security attributes associated with Discretionary Access Control policies on named objects”

This requirement is addressed by the Windows OS as follows.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “7.1.1.1” requirement, to revoke elements of the DACL associated with a named object, the authorized subject must possess the WRITE\_DAC in the DACL.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “2.1.2.1” requirement, WRITE\_DAC is one of the standard access control elements applicable to every named object. It corresponds to the right to modify the DACL of a named object. In other words, it corresponds to the DAC change authorization defined in the Commercial Grade OS Requirement Set. It is a valid access control element which may appear in the access mask of any ACE of a DACL.

If a Windows OS resource manager does not wish to grant WRITE\_DAC to the object creator subject of its object, then it specifies an explicit “creator owner rights” ACE in the DACL of the object, where the “creator owner rights” ACE does not include WRITE\_DAC in its access mask, as explained in the justification text for addressing the Commercial Grade OS Requirement Set “2.1.2.1” requirement.

Typically, Windows OS resource managers grant WRITE\_DAC to authorized administrators by introducing WRITE\_DAC into the access mask for inclusion in an ACCESS\_ALLOWED\_ACE or ACCESS\_ALLOWED\_OBJECT\_ACE ACE, which identifies the “Administrators” local group.

As a result, this Commercial Grade OS Requirement Set “7.1.2.1” requirement is satisfied.

## Addressing 7.1.2.2 “The OS shall allow authorized administrators and subjects with the MIC change attribute the ability to revoke security attributes associated with Mandatory Integrity Control policies on named objects”

This requirement is addressed by the Windows OS as follows.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “7.1.1.1” requirement, to revoke elements of the MACL associated with a named object, the authorized subject must possess the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK).

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “2.2.1.2” requirement, a security descriptor, including a MACL, may be modified by an authorized subject. The authorized subject must possess the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) to request the responsible resource manager to update the MIC security attributes of a specific named object. In other words, the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) corresponds to the MIC change authorization defined in the Commercial Grade OS Requirement Set. The [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) is maintained in the Windows OS authentication service’s policy database.

By default, the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) is assigned to local system only. However, an administrator may assign the [SeRelabelPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) to the “Administrators” group or to a specific user account object.

As a result, this Commercial Grade OS Requirement Set “7.1.2.2” requirement is satisfied.

## Addressing 7.1.2.3 “The OS shall allow authorized administrators the ability to revoke security attributes associated with users”

This requirement is addressed by the Windows OS as follows.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “7.1.1.2” requirement, the Windows OS allows an administrator to revoke the following security attributes associated with a user account:

* [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + By default, a standard user does not possess the right to update this attribute value;
* Passwords security attributes
  + [ATT\_UNICODE\_PWD](http://msdn.microsoft.com/en-us/library/ms680513(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
    - If the user account is domain based, it can be updated by an administrator without the knowledge of the current password as there is the ACCESS\_ALLOWED\_OBJECT\_ACE ACE for the GUID\_CONTROL\_UserForceChangePassword property set to allow the administrator in the user account object’s default security descriptor;
    - If the user account is locally defined, it can be updated by an administrator without the knowledge of the current password as there is the ACCESS\_ALLOWED\_ACE ACE for USER\_FORCE\_PASSWORD\_CHANGE to allow the administrator in the user account object’s default security descriptor;
  + [ATT\_SUPPLEMENTAL\_CREDENTIALS](http://msdn.microsoft.com/en-us/library/ms679920(VS.85).aspx) (if the user account is domain based);
* [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) (if the user account is domain based)
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_USER\_ACCOUNT\_CONTROL](http://msdn.microsoft.com/en-us/library/ms680832(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + Among other code values, it contains specifically the USER\_ACCOUNT\_DISABLED code value for indicating that the specific user account is disabled;
  + By default, a standard user does not possess the right to update this attribute value;
* [ATT\_IS\_MEMBER\_OF\_DL](http://msdn.microsoft.com/en-us/library/ms677099(VS.85).aspx) (if the user account is domain based) or its equivalent for a locally defined user account
  + It contains the distinguished names of the groups to which this user account directly belongs to;
  + Group nesting is possible in Windows OS;
  + A [group object](http://msdn.microsoft.com/en-us/library/ms682251(VS.85).aspx) also has its own ATT\_IS\_MEMBER\_OF\_DL attribute and so SAM is responsible to conduct the group membership expansion;
  + By default, a standard user does not possess the right to update this attribute value;
* the Windows OS privileges and the system access rights maintained in the Windows OS authentication service’s policy database
  + By default, the access control rights (such as [ACCOUNT\_ADJUST\_PRIVILEGES](http://msdn.microsoft.com/en-us/library/ms721750(VS.85).aspx) or [ACCOUNT\_ADJUST\_SYSTEM\_ACCESS](http://msdn.microsoft.com/en-us/library/ms721750(VS.85).aspx)) for an administrator are specified in the security descriptor of a Windows OS LSA policy Account object in the Windows OS authentication service's policy database. The administrator may set or modify the Windows OS privileges or system access for assigning to a Windows OS defined user account or a group containing the user account.

As a result, this Commercial Grade OS Requirement Set “7.1.2.3” requirement is satisfied.

## Addressing 7.1.2.4 “The OS shall allow authorized administrators the ability to delete user accounts”

This requirement is addressed by the Windows OS as follows.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement, the Windows OS defines user account objects in the Active Directory (for Windows OS domain wide user accounts) and in the Windows OS SAM of an individual machine (for Windows OS local user accounts on the machine).

To delete a specified user account object, which is defined locally in the SAM, the SAM [SamrDeleteUser()](http://msdn.microsoft.com/en-us/library/cc212070.aspx) interface is used by the user account object deletion requester. However, the requester must possess the DELETE standard access right. By default, this DELETE standard access right is granted to

* an administrator;
* an account operator.

Additionally, as explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.3.1” requirement, the Windows OS SAM generates the [Event ID 4726](Event%20ID%204767%20(SE_AUDITID_ACCOUNT_UNLOCKED/SE_AUDITID_ETW_ACCOUNT_UNLOCKED)) security audit record “A user account was deleted” (SE\_AUDITID\_USER\_DELETED/ SE\_AUDITID\_ETW\_USER\_DELETED), specifying the deleted user account object, upon the successful completion of a user account object delete request through the SAM [SamrDeleteUser()](http://msdn.microsoft.com/en-us/library/cc212070.aspx) interface.

To delete a specified user account object, which is defined in the Active Directory as a child object in the [Windows OS SAM domain Active Directory object](http://msdn.microsoft.com/en-us/library/ms683940(VS.85).aspx), the Windows OS LDAP server provides the LDAP interface for a subject to attempt the deletion operation using the [LDAP\_DELETE\_CMD](http://msdn.microsoft.com/en-us/library/aa366554(VS.85).aspx) command. The LDAP interface requires the caller subject to possess the RIGHT\_DS\_DELETE\_TREE right in the security descriptor for the user account object. By default, the security descriptor for a user account object grants the RIGHT\_DS\_DELETE\_TREE right to

* an administrator;
* local system;
* an account operator.

Finally, as explained in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.3.1” requirement, the Windows OS Active Directory generates the [Event ID 5141](http://support.microsoft.com/kb/947226) security audit record “A directory service object was deleted” (SE\_AUDITID\_ETW\_DS\_OBJECT\_DELETE\_value), specifying the deleted user account object, upon the successful completion of a user account object delete request through the LDAP interface using the [LDAP\_DELETE\_CMD](http://msdn.microsoft.com/en-us/library/aa366554(VS.85).aspx) command.

As a result, this Commercial Grade OS Requirement Set “7.1.2.4” requirement is satisfied.

# Meeting the “Revocation Access Revocation Audit Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual audit requirement under the heading of “Revocation Access Revocation Audit Requirements”. It is listed as “7.1.3.1”.

## Addressing 7.1.3.1 “The OS shall provide the ability to audit the revocation of security attribute”

This requirement is addressed by the Windows OS as follows.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “7.1.1.2” requirement, the corresponding security audit records are generated when the following security attributes associated with a user account are revoked by an administrator.

* When the [ATT\_LOGON\_HOURS](http://msdn.microsoft.com/en-us/library/ms676846(VS.85).aspx) security attribute is changed successfully by an administrator, the new value is recorded in the “Logon Hours” informational item of the [Event ID 4738](Event%20ID%204767%20(SE_AUDITID_ACCOUNT_UNLOCKED/SE_AUDITID_ETW_ACCOUNT_UNLOCKED)) security audit record “A user account was changed” (SE\_AUDITID\_USER\_CHANGE/ SE\_AUDITID\_ETW\_USER\_CHANGE) as generated by the Windows OS SAM.
* When the [ATT\_UNICODE\_PWD](http://msdn.microsoft.com/en-us/library/ms680513(VS.85).aspx) and the [ATT\_SUPPLEMENTAL\_CREDENTIALS](http://msdn.microsoft.com/en-us/library/ms679920(VS.85).aspx) attributes are updated successfully by an administrator, the [Event ID 4724](http://support.microsoft.com/kb/947226) security audit record “An attempt was made to reset an account's password” (SE\_AUDITID\_USER\_PWD\_SET/SE\_AUDITID\_ETW\_USER\_PWD\_SET) is generated by the Windows OS SAM.
* When the [ATT\_ALT\_SECURITY\_IDENTITIES](http://msdn.microsoft.com/en-us/library/ms675221(VS.85).aspx) attribute is changed successfully by an administrator, the change is recorded in the “Directory Object Attribute Value” informational item of the [Event ID 5136](http://support.microsoft.com/kb/947226) security audit record “A directory service object was modified” (SE\_AUDITID\_ETW\_DS\_OBJECT\_MODIFY\_value) as generated by the Windows OS Active Directory.
* When the USER\_ACCOUNT\_DISABLED code value in the [ATT\_USER\_ACCOUNT\_CONTROL](http://msdn.microsoft.com/en-us/library/ms680832(VS.85).aspx) attribute is changed successfully by an administrator, the change is recorded in the “Old UAC Value”, “New UAC Value” and “User Account Control” informational items of the [Event ID 4738](Event%20ID%204767%20(SE_AUDITID_ACCOUNT_UNLOCKED/SE_AUDITID_ETW_ACCOUNT_UNLOCKED)) security audit record “A user account was changed” (SE\_AUDITID\_USER\_CHANGE/ SE\_AUDITID\_ETW\_USER\_CHANGE) as generated by the Windows OS SAM.
* When the [ATT\_IS\_MEMBER\_OF\_DL](http://msdn.microsoft.com/en-us/library/ms677099(VS.85).aspx) attribute is updated successfully due to a group membership removal revocation action of an administrator, one of the following Event ID security audit records is generated by the Windows OS SAM:
  + [Event ID 4733](http://support.microsoft.com/kb/947226) security audit record “A member was removed from a security-enabled local group” (SE\_AUDITID\_LOCAL\_GROUP\_REM/SE\_AUDITID\_ETW\_LOCAL\_GROUP\_REM), indicating
    - Name of the member removed;
    - ID of the member removed;
  + [Event ID 4729](http://support.microsoft.com/kb/947226) security audit record “A member was removed from a security-enabled global group” (SE\_AUDITID\_GLOBAL\_GROUP\_REM/SE\_AUDITID\_ETW\_GLOBAL\_GROUP\_REM), indicating
    - Name of the member removed;
    - ID of the member removed;
  + [Event ID 4757](http://support.microsoft.com/kb/947226) security audit record “A member was removed from a security-enabled universal group” (SE\_AUDITID\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_REM/ SE\_AUDITID\_ETW\_SECURITY\_ENABLED\_UNIVERSAL\_GROUP\_REM), indicating
    - Name of the member removed;
    - ID of the member removed.
* The policy database component of the Windows OS authentication service is responsible to generate the following security audit records when the Windows OS privilege (i.e. user right) and system access assignment for a user account or a group is revoked by an administrator:
  + [Event ID 4705](http://support.microsoft.com/kb/947226) security audit record “A user right was removed” (SE\_AUDITID\_ETW\_USER\_RIGHT\_REMOVED\_value)
    - Specific rights (i.e. Windows OS privileges) removed;
  + [Event ID 4718](http://support.microsoft.com/kb/947226) security audit record “System security access was removed from an account” (SE\_AUDITID\_ETW\_SYSTEM\_ACCESS\_REMOVED\_value)
    - Specific security access removed.

As a result, this Commercial Grade OS Requirement Set “7.1.3.1” requirement is satisfied.

# Meeting the “Auditing Audit Collection Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 4 individual functional requirements under the heading of “Auditing Audit Collection Functional Requirements”. They are listed as “8.1.1.n”, where n = 1, 2, 3 and 4.

## Addressing 8.1.1.1 “The OS shall be able to generate audit records for all security-relevant events identified in this Commercial Grade OS Requirement Set and the other specific security relevant auditable events designed to be generated by the OS claiming compliance with this Commercial Grade OS Requirement Set”

The individual audit event records being generated are described in the justification text of their corresponding applicable functional or management requirements which specify the exact conditions under which these audit event records are generated. Hence, there is no need to repeat them here.

## Addressing 8.1.1.2 “The OS shall be able to associate each auditable event with the identity or the user that caused the event”

The individual audit event records being generated are described in the justification text of their corresponding applicable functional or management requirements which specify the exact conditions under which these audit event records are generated. Where appropriate for a specific event, the identity that caused the specific event is mentioned as an information item for inclusion in the corresponding event record(s). Hence, there is no need to repeat them here.

## Addressing 8.1.1.3 “The OS shall be able to monitor and report the accumulation of specific sets of audit events known to indicate a potential security violation and immediately report the accumulated events when a threshold is exceeded”

The Commercial Grade OS Requirement Set requires the accumulation of the following events for reporting:

1. an administrator specified number of individual user authentication failures within an administrator specified time period;
2. an administrator specified number of Discretionary Access Control policy violation attempts by an individual user within an administrator specified time period;
3. an administrator specified number of Mandatory Integrity Control policy violation attempts by an individual user within an administrator specified time period;
4. any self-tests failure.

This requirement is addressed by Microsoft as follows.

We believe that the monitoring and reporting capabilities based on the operating system generated security relevant audit event records are not a unique “security” feature of an operating system. Other information assurance (IA) and information assurance enabled products also generate their corresponding security relevant audit event records. While the underlying meanings of audit event records generated from various IA products and IA enabled products are different, the actual monitoring activities of the generated event records and the subsequent reporting infrastructures for realizing the monitoring and reporting capabilities of the products need not be different. The enterprise administrator community actually would argue that they prefer

* a single mechanism to monitor event records generated from all deployed products/systems within their distributed environments;
* a single infrastructure to report to them through a means (such as a dialog popping up on their working workstations, email, instant message, cell phone, or others) that they would want to select.

The [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) addresses these needs of the enterprise administrator community by providing end-to-end monitoring for an enterprise IT environment. The [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) can monitor thousands of servers, applications, and client machines and provides comprehensive views of their health states. These views are the key to a rapid and agile response to events that can impact the availability and security of services that the enterprise administrators provide to their users. The [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) is also capable to collect and to respond to UNIX/Linux Syslog messages as described in the [KB942863](http://support.microsoft.com/kb/942863) article.

In the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx), [Audit Collection Services (ACS)](http://technet.microsoft.com/en-us/library/bb381258.aspx) provides a means to collect audit event records generated, due to the Windows OS audit policy currently being enforced, and to store them in a centralized database. By default, when the Windows OS audit policy is enforced in a Windows OS machine, that Windows OS machine automatically saves all audit event records, generated due to the audit policy, to its local hard audit store. Using ACS, the administrators consolidate individual hard audit stores into a centrally managed database. They filter and analyze the collected audit event records using the data analysis and reporting tools provided by the Microsoft SQL Server. With ACS, only a user who specifically has been given the necessary right to access the ACS database can run queries and create reports on the collected data. In the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx), the deployment of ACS involves the following components:

* ACS forwarders;
* ACS collector;
* ACS database.

An ACS forwarder runs on an individual managed machine within the administrators’ enterprise environment. By default, the ACS forwarder is installed but not enabled when the Operations Manager agent is installed as part of the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) deployment. After the ACS forwarder is enabled through the “[Enable Audit Collection](http://technet.microsoft.com/en-us/library/bb381332.aspx)” task of the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx), all audit event records are sent to the ACS collector while the local [Windows OS security audit store service (wevtsvc.dll)](http://msdn.microsoft.com/en-us/library/cc204960.aspx) is saving them in the local hard audit store.

The ACS collector receives and processes audit event records from ACS forwarders and then sends this data to the ACS database. This processing includes disassembling the data so that it can be spread across several tables within the ACS database, minimizing data redundancy, and also applying filters so that unnecessary audit event record information items are not added to the ACS database.

The ACS database is the central repository for audit event records that have been generated due to the Windows OS audit policies enforced in the managed Windows OS machines within an ACS deployment of the administrators’ enterprise environment. The ACS database can be located on the same Windows OS machine as the ACS collector, but for best performance, each may also be installed on a dedicated Windows OS server machine.

The [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) provides the capability to monitor the occurrences of correlated audit event records as implied by this Commercial Grade OS Requirement Set “8.1.1.3” requirement. This capability is explained further in the [Microsoft Operations Manager 2007 online help](http://technet.microsoft.com/en-us/library/bb381298.aspx).

Through its concept of “notification channel”, the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) also provides the capability to report the occurrences of monitored audit event records as implied by this Commercial Grade OS Requirement Set “8.1.1.3” requirement. This capability is explained further in the [Microsoft Operations Manager 2007 online help](http://technet.microsoft.com/en-us/library/bb381452.aspx).

As a result, this Commercial Grade OS Requirement Set “8.1.1.3” requirement is met by the capabilities of the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) as long as the specific audit event records listed in a) to d) above are generated by the Windows OS and are written to the hard audit store on an individual Windows OS machine.

## Addressing 8.1.1.4 “The OS shall record, within each audit record, the specific set of information items, as appropriate to the audit event”

The Commercial Grade OS Requirement Set requires the following set of information items to be recorded within an audit event record, as appropriate to the corresponding audit event:

1. date, time, and location of the event;
2. type of event;
3. event outcome (success or failure);
4. name of the object;
5. old and new values (except for authentication data and critical cryptographic security parameters) of changed security relevant data.

This requirement is addressed by the Windows OS as follows.

The individual audit event records being generated are described in the justification text of their corresponding applicable functional or management requirements which specify the exact conditions under which these audit event records are generated.

The date and time of the event are always included as an information item in every event record.

Where appropriate for a specific event, the location in the form of the unauthenticated IP address where the client resides is included as an information item in the corresponding event record(s).

The type of an event is indicated by the “Category” or the “Subcategory” field of the corresponding event record.

Where appropriate for a specific event, the success or failure outcome is included as an information item in the corresponding event record.

Where appropriate for a specific event, the name of the object associated with the event is included as an information item in the corresponding event record(s).

Where appropriate for a specific event, the old and new values for a configuration element associated with the event are included as an information item in the corresponding event record.

Hence, there is no need to repeat all individual audit event record descriptions here.

# Meeting the “Auditing Audit Collection Management Requirements”

In the Commercial Grade OS Requirement Set, there are 2 individual management requirements under the heading of “Auditing Audit Collection Management Requirements”. They are listed as “8.1.2.n”, where n = 1 and 2.

## Addressing 8.1.2.1 “The OS shall provide an authorized administrator with the capability to manage the threshold values specified in “8.1.1.3””

This requirement is addressed by Microsoft as follows.

We have suggested the use of the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) to address the Commercial Grade OS Requirement Set “8.1.1.3” requirement.

The administrator specified “time period” threshold value specified in the “8.1.1.3” requirement is interpreted as the “[Correlation Interval](http://technet.microsoft.com/en-us/library/bb381298.aspx)” value which is configured when the monitoring of a set of correlated audit event records is defined, using the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx).

The administrator specified “number of individual user authentication failures” threshold value specified in the “8.1.1.3” requirement is interpreted as the number of times where the [Event ID 4625](http://support.microsoft.com/kb/947226) “An account failed to log on” (SE\_AUDITID\_ETW\_LOGON\_FAILURE\_value) for the specific individual user account is specified in the set of correlated audit event records for monitoring, using the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) .

The administrator specified “number of Discretionary Access Control Policy violation attempts by an individual user” threshold value specified in the “8.1.1.3” requirement is interpreted as the number of times where

* the failure [Event ID 4656](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OPEN\_HANDLE\_value) “A handle to an object was requested” for the specific individual user account; or
* the failure [Event ID 4662](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OBJECT\_OPERATION\_value) “An operation was performed on an object” for the specific individual user account

is specified in the set of correlated audit event records for monitoring, using the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx).

The administrator specified “number of Mandatory Integrity Control Policy violation attempts by an individual user” threshold value specified in the “8.1.1.3” requirement is interpreted as the number of times where the failure [Event ID 4656](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OPEN\_HANDLE\_value) “A handle to an object was requested” for the specific individual user account is specified in the set of correlated audit event records for monitoring, using the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx).

Note that the enforcement of the Mandatory Integrity Control Policy is exercised before the enforcement of the Discretionary Access Control Policy. Therefore, if the enforcement of the Mandatory Integrity Control Policy results in a denial of the access request, there is no need to exercise the enforcement of the Discretionary Access Control Policy.

As explained in the justification text for addressing the Commercial Grade OS Requirement Set “4.5.1.1” requirement, the Windows OS generates [Event ID 5056](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_NCRYPT\_SELF\_TEST\_EVENT\_value) “A cryptographic self test was performed” for the success or failure outcome resulted from the cryptographic self tests of the FIPS 140-2 validated [Windows OS Cryptographic Primitives Library](http://msdn.microsoft.com/en-us/library/aa833130(VS.85).aspx) (bcrypt.dll) when it is loaded into the [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) (keyiso.dll) through the wrapper (ncrypt.dll). The [Windows OS CNG key isolation service](http://msdn.microsoft.com/en-us/library/bb204778.aspx) is hosted in the LSASS.exe Windows OS process (i.e. the same process that also hosts the Windows OS Authentication Service). The service provides cryptographic key process isolation to cryptographic public / private key pairs and associated cryptographic operations.

## Addressing 8.1.2.2 “The OS shall allow an authorized administrator to specify which events, from the set of auditable events, are to be audited”

This requirement is addressed by the Windows OS as follows.

The audit policy for the local Windows OS machine is managed and maintained by the Windows OS audit policy unit residing within the LSASS.exe Windows OS process (i.e. the same process that also hosts the Windows OS Authentication Service). The Windows OS audit policy unit provides a local RPC interface [LsarSetAuditPolicy()](http://msdn.microsoft.com/en-us/library/aa375712(VS.85).aspx)[[30]](#footnote-31) for an authorized subject to define specific elements of the audit policy that is enforced within the local Windows OS machine.

The audit policy defines a specific security event type (in terms of an audit subcategory within an audit category) and decides whether audit records of that type needs to be generated or not. The supported audit categories and audit subcategories are listed as follows with their default audit policy elements.

* [Audit\_System](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
  + [Audit\_System\_SecurityStateChange](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events;
  + [Audit\_System\_SecuritySubsystemExtension](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_System\_Integrity](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success and failure events;
  + [Audit\_System\_IPSecDriverEvents](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_System\_Others](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success and failure events;
* [Audit\_Logon](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
  + [Audit\_Logon\_Logon](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success and failure events in Windows Server 2008 machines;
    - By default, audit records are generated for their success events in Windows Vista machines;
  + [Audit\_Logon\_Logoff](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events;
  + [Audit\_Logon\_AccountLockout](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events;
  + [Audit\_Logon\_IPSecMainMode](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_Logon\_IPSecQuickMode](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_Logon\_IPSecUserMode](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_Logon\_SpecialLogon](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events;
  + [Audit\_Logon\_Others](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_Logon\_NPS](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success and failure events;
* [Audit\_ObjectAccess](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
  + [Audit\_ObjectAccess\_FileSystem](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_ObjectAccess\_Registry](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_ObjectAccess\_Kernel](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_ObjectAccess\_Sam](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_ObjectAccess\_CertificationServices](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_ObjectAccess\_ApplicationGenerated](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_ObjectAccess\_Handle](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_ObjectAccess\_Share](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_ObjectAccess\_FirewallPacketDrops](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_ObjectAccess\_FirewallConnection](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_ObjectAccess\_Other](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
* [Audit\_PrivilegeUse](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
  + [Audit\_PrivilegeUse\_Sensitive](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_PrivilegeUse\_NonSensitive](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_PrivilegeUse\_Others](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
* [Audit\_DetailedTracking](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
  + [Audit\_DetailedTracking\_ProcessCreation](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_DetailedTracking\_ProcessTermination](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_DetailedTracking\_DpapiActivity](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_DetailedTracking\_RpcCall](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
* [Audit\_PolicyChange](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
  + [Audit\_PolicyChange\_AuditPolicy](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events;
  + [Audit\_PolicyChange\_AuthenticationPolicy](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events;
  + [Audit\_PolicyChange\_MpsscvRulePolicy](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_PolicyChange\_WfpIPSecPolicy](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_PolicyChange\_Others](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
* [Audit\_AccountManagement](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
  + [Audit\_AccountManagement\_UserAccount](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events;
  + [Audit\_AccountManagement\_ComputerAccount](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events;
  + [Audit\_AccountManagement\_SecurityGroup](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events;
  + [Audit\_AccountManagement\_DistributionGroup](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_AccountManagement\_ApplicationGroup](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_AccountManagement\_Others](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
* [Audit\_DirectoryServiceAccess](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
  + [Audit\_DSAccess\_DSAccess](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events in Windows Server 2008 Domain Controllers;
  + [Audit\_DSAccess\_AdAuditChanges](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_Ds\_Replication](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_Ds\_DetailedReplication](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
* [Audit\_AccountLogon](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
  + [Audit\_AccountLogon\_CredentialValidation](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events in Windows Server 2008 Domain Controllers;
  + [Audit\_AccountLogon\_Kerberos](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events in Windows Server 2008 Domain Controllers;
  + [Audit\_AccountLogon\_Others](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx);
    - By default, no audit records are generated;
  + [Audit\_AccountLogon\_KerbCredentialValidation](http://msdn.microsoft.com/en-us/library/bb648638(VS.85).aspx)
    - By default, audit records are generated for their success events in Windows Server 2008 Domain Controllers.

For each audit category or audit subcategory, the following instructions can be applied to its audit events for the audit event record generation effects of these audit events.

* [POLICY\_AUDIT\_EVENT\_UNCHANGED](http://msdn.microsoft.com/en-us/library/aa965467(VS.85).aspx);
* [POLICY\_AUDIT\_EVENT\_SUCCESS](http://msdn.microsoft.com/en-us/library/aa965467(VS.85).aspx);
* [POLICY\_AUDIT\_EVENT\_FAILURE](http://msdn.microsoft.com/en-us/library/aa965467(VS.85).aspx);
* [POLICY\_AUDIT\_EVENT\_NONE](http://msdn.microsoft.com/en-us/library/aa965467(VS.85).aspx).

For each audit category or audit subcategory, the following instructions can be applied to its audit events for the audit event record generation effects of these audit events on the per user basis with a target user account SID specified in the [LsarSetAuditPolicy()](http://msdn.microsoft.com/en-us/library/aa375706(VS.85).aspx) interface.

* [PER\_USER\_POLICY\_UNCHANGED](http://msdn.microsoft.com/en-us/library/aa965467(VS.85).aspx);
* [PER\_USER\_AUDIT\_SUCCESS\_INCLUDE](http://msdn.microsoft.com/en-us/library/aa965467(VS.85).aspx);
* [PER\_USER\_AUDIT\_SUCCESS\_EXCLUDE](http://msdn.microsoft.com/en-us/library/aa965467(VS.85).aspx);
* [PER\_USER\_AUDIT\_FAILURE\_INCLUDE](http://msdn.microsoft.com/en-us/library/aa965467(VS.85).aspx);
* [PER\_USER\_AUDIT\_FAILURE\_EXCLUDE](http://msdn.microsoft.com/en-us/library/aa965467(VS.85).aspx);
* [PER\_USER\_AUDIT\_NONE](http://msdn.microsoft.com/en-us/library/aa965467(VS.85).aspx).

The audit policy resulted from the [LsarSetAuditPolicy()](http://msdn.microsoft.com/en-us/library/aa375706(VS.85).aspx) interface are stored in the policy database maintained by the LSASS.exe Windows OS process (i.e. the same process that also hosts the Windows OS Authentication Service).

To use the [LsarSetAuditPolicy()](http://msdn.microsoft.com/en-us/library/aa375706(VS.85).aspx) interface to configure an element of the audit policy, the caller subject must possess either the [SeSecurityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) or the [AUDIT\_SET\_SYSTEM\_POLICY](http://msdn.microsoft.com/en-us/library/aa375712(VS.85).aspx) permission right or the [AUDIT\_SET\_USER\_POLICY](http://msdn.microsoft.com/en-us/library/aa375706(VS.85).aspx) permission right. By default, the [SeSecurityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) is enabled for any member of the “Administrators” local group. However, neither the AUDIT\_SET\_SYSTEM\_POLICY permission right nor the AUDIT\_SET\_USER\_POLICY permission right is assigned to any user account by default.

If an element of the audit policy is configured successfully through the LsarSetAuditPolicy() interface, then the Windows OS audit policy unit generates either the [Event ID 4912](http://support.microsoft.com/kb/947226) or [Event ID 4719](http://support.microsoft.com/kb/947226) security audit record.

* [Event ID 4912](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_PER\_USER\_POLICY\_CHANGE\_value) “Per User Audit Policy was changed” with the following informational items when available:
  + Subject SID:
  + Subject User Name:
  + Subject Domain Name:
  + Subject Logon ID:
  + Target User SID:
  + Audit Category:
  + Audit Subcategory:
  + Audit Policy Changes: (which describe the specific element changes of the audit policy).
* [Event ID 4719](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_POLICY\_CHANGE\_BY\_SUBCATEGORY\_value) “System audit policy was changed” with the following informational items when available:
  + Subject SID:
  + Subject User Name:
  + Subject Domain Name:
  + Subject Logon ID:
  + Audit Category:
  + Audit Subcategory:
  + Audit Policy Changes: (which describe the specific element changes of the audit policy).

Assigning the [AUDIT\_SET\_SYSTEM\_POLICY](http://msdn.microsoft.com/en-us/library/aa375712(VS.85).aspx) permission right or the [AUDIT\_SET\_USER\_POLICY](http://msdn.microsoft.com/en-us/library/aa375706(VS.85).aspx) permission right to a user account is possible within the Windows OS. This can be achieved by an authorized subject possessing the [SeSecurityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) using the [LsarSetAuditSecurity()](http://msdn.microsoft.com/en-us/library/aa375706(VS.85).aspx) provided by the Windows OS audit policy unit. If the assignment is successful, then the Windows OS audit policy unit generates the [Event ID 4715](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_SECURITY\_OBJECT\_CHANGE\_value) “The audit policy (SACL) on an object was changed” with the following informational items when available:

* Subject SID:
* Subject User Name:
* Subject Domain Name:
* Subject Logon ID:
* Old Security Descriptor:
* New Security Descriptor:.

Finally, as mentioned in the justification text for addressing the Commercial Grade OS Requirement Set “2.1.3.1” requirement, the [system access control list (SACL)](http://msdn.microsoft.com/en-us/library/aa375723(VS.85).aspx) of [SYSTEM\_AUDIT\_ACE](http://msdn.microsoft.com/en-us/library/aa379616(VS.85).aspx) ACEs and [SYSTEM\_AUDIT\_OBJECT\_ACE](http://msdn.microsoft.com/en-us/library/aa379619(VS.85).aspx) ACEs within a [security descriptor structure](http://msdn2.microsoft.com/en-us/library/aa379561(VS.85).aspx) is the mechanism that the Windows OS uses for associating the auditing requirements of the DAC policy decisions with a named object. Modifying the SACL associated with a named object is possible within the Windows OS. This can be achieved by an authorized subject possessing the [SeSecurityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK). If the modification is successful, then the [Event ID 4907](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_SACL\_CHANGE\_value) “Auditing settings on object were changed” with the following informational items when available:

* Subject:
* Security ID:
* Account Name:
* Account Domain:
* Logon ID:
* Object:
* Object Server:
* Object Type:
* Object Name:
* Handle ID:
* Process:
* Process ID:
* Process Name:
* Original Security Descriptor:
* New Security Descriptor:.

As a result, this Commercial Grade OS Requirement Set “8.1.2.2” requirement is met.

# Meeting the “Auditing Audit Collection Audit Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual audit requirement under the heading of “Auditing Audit Collection Audit Requirements”. It is listed as “8.1.3.1”.

## Addressing 8.1.3.1 “The OS shall provide the ability to audit the modification of the threshold values specified in “8.1.1.3””

This requirement is addressed by Microsoft as follows.

As mentioned in the justification text for addressing the Commercial Grade OS Requirement Set “8.1.1.3” requirement, the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) provides the capability to monitor the occurrences of correlated audit event records as implied by the Commercial Grade OS Requirement Set “8.1.1.3” requirement. Each set of correlated audit event records for monitoring is managed and maintained for perseverance by the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx). An authorized user can use the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) to view specific aspects of the sets of correlated audit event records for monitoring. These aspects include:

* the “[Correlation Interval](http://technet.microsoft.com/en-us/library/bb381298.aspx)” value which is interpreted as the administrator specified “time period” threshold value specified in the “8.1.1.3” requirement;
* the number of times where the [Event ID 4625](http://support.microsoft.com/kb/947226) “An account failed to log on” (SE\_AUDITID\_ETW\_LOGON\_FAILURE\_value) for the specific individual user account is specified in the corresponding set of correlated audit event records for monitoring, and this number is interpreted as the administrator specified “number of individual user authentication failures” threshold value specified in the “8.1.1.3” requirement;
* the number of times where
  + the failure [Event ID 4656](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OPEN\_HANDLE\_value) “A handle to an object was requested” for the specific individual user account; or
  + the failure [Event ID 4662](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_OBJECT\_OPERATION\_value) “An operation was performed on an object” for the specific individual user account

is specified in the corresponding set of correlated audit event records for monitoring, and this number is interpreted as the administrator specified “number of Discretionary Access Control Policy violation attempts by an individual user” threshold value and the administrator specified “number of Mandatory Integrity Control Policy violation attempts by an individual user” threshold value, specified in the “8.1.1.3” requirement.

As a result, due to the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) capabilities, an explicit audit record for auditing the modification of the threshold values specified in the “8.1.1.3” requirement is deemed as unnecessary.

# Meeting the “Auditing Audit Storage Functional Requirements”

In the Commercial Grade OS Requirement Set, there are 2 individual functional requirements under the heading of “Auditing Audit Storage Functional Requirements”. They are listed as “8.2.1.n”, where n = 1, and 2.

## Addressing 8.2.1.1 “The OS shall prevent modification of previously written audit records”

This requirement is addressed by the Windows OS as follows.

### Windows OS audit policy unit

Due to the needed enforcement of the audit policy, all generated audit event records from the various responsible audit generating modules within the Windows OS need to be queued and then filtered by the Windows OS audit policy unit.

When the LSASS.exe Windows OS process starts, the Windows OS audit policy unit is initialized. During its initialization, the Windows OS audit policy unit calls the kernel mode [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx)’s “Trace Control” interface with the EtwRegisterSecurityProv function code to register itself as the security provider with the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx). The effect of this registration is to let the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) remember the Windows OS audit policy unit as the only subject that is allowed to trace (i.e. write) an audit event record through the [NtTraceEvent()](http://msdn.microsoft.com/en-us/library/aa364137.aspx) interface of the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) with the “ETW\_NT\_FLAGS\_TRACE\_SECURITY” flag. The [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) traces the submitted audit event records in the [ETW logger](http://msdn.microsoft.com/en-us/library/aa363694(VS.85).aspx), named “EventLog-Security” until the records are consumed by the logger’s authorized consumers. An ETW logger is an [ETW trace session](http://msdn.microsoft.com/en-us/library/aa363694(VS.85).aspx) that records events from an event trace provider. Therefore, the Windows OS audit policy unit is the only event trace provider for the “EventLog-Security” logger.

As the Windows OS audit policy unit is the only event trace provider for the “EventLog-Security” logger, an audit event record cannot be traced into the “EventLog-Security” logger, unless the audit event record has been mediated by the Windows OS audit policy unit. Responsible audit generating modules within the Windows OS that wish to generate an audit event record have to use the appropriate interfaces (e.g. [AuthzReportSecurityEvent()](http://msdn.microsoft.com/en-us/library/aa376317(VS.85).aspx) in the user mode) provided by the Windows OS audit policy unit to queue their generated audit event records for being traced into the “EventLog-Security” logger, subject to the current audit policy enforcement. The corresponding Windows OS audit policy unit interfaces require that their caller subjects possess the necessary [SeAuditPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK). As mentioned in the introduction of [SeAuditPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) earlier in this paper, the [SeAuditPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK), by default, is assigned to the local system, local service, and network service.

### Windows OS event tracing (ETW) facility

As the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) is part of the Windows OS kernel execution module (e.g. ntoskrnl.exe), it is initialized as the Windows OS machine boots. During its initialization, the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) starts the well known auto loggers based on the information stored in the following [registry key](http://msdn.microsoft.com/en-us/library/aa363687(VS.85).aspx).

* HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\WMI\Autologger.

One of the well known auto loggers that the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) starts automatically during boot is the “EventLog-Security” logger due to the “EventLog-Security” registry subkey. The “EventLog-Security” logger GUID is “0e66e20b-b802-ba6a-9272-31199d0ed295”. In addition, the “EventLog-Security” logger is started with the following log modes according to the “LogFileMode” registry key value:

* [EVENT\_TRACE\_REAL\_TIME\_MODE (0x00000100)](http://msdn.microsoft.com/en-us/library/aa364080(VS.85).aspx);
* [EVENT\_TRACE\_NONSTOPPABLE\_MODE (0x00000040)](http://msdn.microsoft.com/en-us/library/aa364080(VS.85).aspx);
* [EVENT\_TRACE\_SECURE\_MODE (0x00000080)](http://msdn.microsoft.com/en-us/library/aa364080(VS.85).aspx).

As the EVENT\_TRACE\_REAL\_TIME\_MODE is configured for the “EventLog-Security” logger, the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) delivers the traced audit event records to the “EventLog-Security” logger’s authorized consumers in real time, after one of the authorized consumers informs the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) that it wishes to process the traced audit event records from the “EventLog-Security” logger.

After the start of the “EventLog-Security” logger, the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) maintains a LoggerContext for the logger. The LoggerContext for the “EventLog-Security” logger includes

* the LoggerId of “AUDIT\_LOGGER\_ID” (2);
* the LoggerName of “EventLog-Security”;
* the InstanceGuid of AuditLoggerGuid (“0e66e20b-b802-ba6a-9272-31199d0ed295”).

The [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) allows only one LoggerContext for the “EventLog-Security” logger. If there is another thread attempting to start the “EventLog-Security” logger, an error status of “[STATUS\_OBJECT\_NAME\_COLLISION](http://msdn.microsoft.com/en-us/library/cc704588.aspx)” is returned to that other thread.

When the Windows OS audit policy unit traces (i.e. writes) an audit event record through the [NtTraceEvent()](http://msdn.microsoft.com/en-us/library/aa364137.aspx) interface with the “ETW\_NT\_FLAGS\_TRACE\_SECURITY” flag, the audit event record is associated with the SecurityProviderGuid (“54849625-5478-4994-a5ba-3e3b0328c30d”) (which becomes the audit event record’s “[EventHeader->ProviderId](http://msdn.microsoft.com/en-us/library/aa363759(VS.85).aspx)”) and the LoggerId of “AUDIT\_LOGGER\_ID” (2). Due to the LoggerContext for the “EventLog-Security” logger, the LoggerId associated with the audit event record allows only the audit event record to be buffered into the “EventLog-Security” logger.

A registry subkey entry of the “54849625-5478-4994-a5ba-3e3b0328c30d” (SecurityProviderGuid) resides in the following registry key:

* HKEY\_LOCAL\_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\WINEVT\Publishers.

This registry subkey specifies the locations of the UI resources (i.e. human readable text in different languages) which would be needed when the audit event records are displayed in an event viewer application.

The [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) automatically registers the SecurityProviderGuid (“54849625-5478-4994-a5ba-3e3b0328c30d”) in its internal “Security Provider Guid Entry” variable and binds the SecurityProviderGuid to the LoggerId of “AUDIT\_LOGGER\_ID” (2) during the Windows OS event tracing (ETW) facility’s initialization. Therefore, there is no need for an external request to register the SecurityProviderGuid (“54849625-5478-4994-a5ba-3e3b0328c30d”).

If a subject attempts to register the SecurityProviderGuid (“54849625-5478-4994-a5ba-3e3b0328c30d”) through the use of the EtwRegisterGuidsCode function code in the “Trace Control” interface, then an error status of “[STATUS\_ACCESS\_DENIED](http://msdn.microsoft.com/en-us/library/cc704588.aspx)” is returned.

### “Etw Logger” Windows OS system thread

There is a unique “Etw Logger” Windows OS system thread running in the security context of the local system. It manages and maintains the LoggerContext for the “EventLog-Security” logger. The “Etw Logger” thread monitors the state of the “EventLog-Security” logger LoggerContext every second[[31]](#footnote-32) to determine if there is any traced audit event record which has been buffered into the “EventLog-Security” logger for delivering to the “EventLog-Security” logger’s authorized consumers. The “Etw Logger” thread creates a persistent file, which is used for saving real-time buffered audit event records that could not be delivered to the authorized consumers for any reason. This file is named “EtwRTEventLog-Security.etl” residing in the local “$SystemRoot$\system32\Logfiles\WMI\RtBackup” directory. The security descriptor for the directory allows only the local system to have any access to its children objects (i.e. the SDDL format of “D:P(A;OICI;FA;;;SY)”). Therefore, the “EtwRTEventLog-Security.etl” does not allow any subject other than the local system to have the write access.

### Windows OS security audit store service

By default, the “EventLog-Security” logger has at least one authorized consumer. This authorized consumer is the local [Windows OS security audit store service (wevtsvc.dll)](http://msdn.microsoft.com/en-us/library/cc204960.aspx). The Windows OS security audit store service resides in a Windows OS process running in the security context of the local service. This service additionally has its own service SID, which is “S-1-5-80-880578595-1860270145-482643319-2788375705-1540778122”.

The Windows OS security audit store service (wevtsvc.dll) has the [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx) object where its internal “security Log” member variable is set as TRUE. The properties of the “Security” channel are stored in the following registry key:

* HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security.

The [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx) has the [ETW session](http://msdn.microsoft.com/en-us/library/aa363694(VS.85).aspx), named “Eventlog-Security”. This “Eventlog-Security” name is set in the [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx)’s internal “Etw Session For Channel session Name” member variable. The “Eventlog-Security” ETW session also has the session ID being the “EventLog-Security” logger GUID (“0e66e20b-b802-ba6a-9272-31199d0ed295”). It is set in the internal “Etw Session For Channel session Id” member variable.

To establish the “Eventlog-Security” ETW session with the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) for the [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx), the Windows OS security audit store service calls [OpenTrace()](http://msdn.microsoft.com/en-us/library/aa364089(VS.85).aspx) to submit an [EVENT\_TRACE\_LOGFILE](http://msdn.microsoft.com/en-us/library/aa363780(VS.85).aspx) structure. This submitted [EVENT\_TRACE\_LOGFILE](http://msdn.microsoft.com/en-us/library/aa363780(VS.85).aspx) structure includes

* the “LoggerName” field, which is the ETW session name “Eventlog-Security”;
* the “[EventRecordCallback](http://msdn.microsoft.com/en-us/library/aa363743(VS.85).aspx)” field, which is the internal “Publisher Manager Event Callback” function used by the Windows OS security audit store service to process the audit event records delivered by the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx);
* the “ProcessTraceMode/LogFileMode” field, which includes the PROCESS\_TRACE\_MODE\_REAL\_TIME/EVENT\_TRACE\_REAL\_TIME\_MODE flag.

### “Eventlog-Security” ETW session establishment

After calling [OpenTrace()](http://msdn.microsoft.com/en-us/library/aa364089(VS.85).aspx), [ProcessTrace()](http://msdn.microsoft.com/en-us/library/aa364093(VS.85).aspx) is called to complete the “Eventlog-Security” ETW session establishment, which includes:

* the sending of the EtwRealtimeConnectCode function code to the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) with the LoggerId of “AUDIT\_LOGGER\_ID” (2) in the “Trace Control” interface;
* the security check that this EtwRealtimeConnectCode sender subject possesses the TRACELOG\_ACCESS\_REALTIME access right for becoming an authorized real time consumer of the “EventLog-Security” logger, according to the security descriptor associated with the AuditLoggerGuid (“0e66e20b-b802-ba6a-9272-31199d0ed295”) residing in the following registry key:
  + HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\WMI\Security;
* the creation of a Windows OS namedpipe (of the null name) residing in the local “\Device\NamedPipe\” directory for the “EventLog-Security” logger “Etw Logger” thread to communicate with this EtwRealtimeConnectCode sender subject;
* notifying the “EventLog-Security” logger “Etw Logger” thread of this ETW session establishment for this EtwRealtimeConnectCode sender subject as an authorized real time consumer of the “EventLog-Security” logger;
* the returning of a user mode handle to the Windows OS namedpipe to this EtwRealtimeConnectCode sender subject.

By default, the Windows OS security audit store service (S-1-5-80-880578595-1860270145-482643319-2788375705-1540778122) is granted the TRACELOG\_ACCESS\_REALTIME access right in the security descriptor associated with the AuditLoggerGuid (“0e66e20b-b802-ba6a-9272-31199d0ed295”).

As the Windows OS security audit store service is the EtwRealtimeConnectCode sender subject, it possesses the handle to the Windows OS namedpipe that the “EventLog-Security” logger “Etw Logger” thread uses to communicate with its authorized real time consumers. To deliver a buffered audit event record to the Windows OS security audit store service, the “EventLog-Security” logger “Etw Logger” thread writes the buffered audit event record into the location of a free virtual memory belonging to the Windows OS process hosting the Windows OS security audit store service. The “EventLog-Security” logger “Etw Logger” thread then informs the Windows OS security audit store service about the address of the written virtual memory location through the Windows OS namedpipe. After reading the address from the Windows OS namedpipe, the buffered audit event record residing at the virtual memory location is mapped into an [EVENT\_RECORD](http://msdn.microsoft.com/en-us/library/aa363769(VS.85).aspx) structure. The mapped [EVENT\_RECORD](http://msdn.microsoft.com/en-us/library/aa363769(VS.85).aspx) structure is then passed to the Windows OS security audit store service’s “[EventRecordCallback](http://msdn.microsoft.com/en-us/library/aa363743(VS.85).aspx)” function associated with the [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx) for further processing.

### Windows OS audit policy unit submitting audit records directly to the Windows OS security audit store service “Security” channel

There are two events for which their Windows OS audit policy unit generated audit event records are submitted directly to the Windows OS security audit store service [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx) without going through the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx). They are described as follows.

* Event ID 521 (SE\_AUDITID\_UNABLE\_TO\_LOG\_EVENTS) “Unable to log events to security log” indicating the failure problem, in terms of the status code, and the number of failed audits
  + It is generated due to the administrator specified “CrashOnAuditFail” policy when
    - the Windows OS audit policy unit is unable to trace (i.e. write) a generated audit event record through the [NtTraceEvent()](http://msdn.microsoft.com/en-us/library/aa364137.aspx) interface of the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) with the “ETW\_NT\_FLAGS\_TRACE\_SECURITY” flag;
    - the Windows OS audit policy unit has to crash the Windows OS machine due to the administrator specified “CrashOnAuditFail” policy.
* Event ID 525 (SE\_AUDITID\_CRASHONAUDITFAIL\_RECOVERY) “Administrator recovered system from CrashOnAuditFail. LSA will now accept non-administrative logons. Some auditable activity might not have been recorded.”
  + It is generated after an administrator
    - has rebooted the Windows OS machine after the machine crash because of the administrator specified “CrashOnAuditFail” policy;
    - has successfully logged on the Windows OS machine;
    - has successfully cleared the security log file;
    - has reconfigured the “CrashOnAuditFail” policy.

Note that an administrator can configure the “CrashOnAuditFail” policy through the “CrashOnAuditFail” registry key value under the following registry key.

* HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa.

If the “CrashOnAuditFail” policy is set, then the Windows OS audit policy unit crashes the Windows OS machine to produce [STATUS\_AUDIT\_FAILED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) (0xC0000244L) on the blue screen when it is unable to trace (i.e. write) a generated audit event record through the [NtTraceEvent()](http://msdn.microsoft.com/en-us/library/aa364137.aspx) interface of the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) with the “ETW\_NT\_FLAGS\_TRACE\_SECURITY” flag. A full security log file (i.e. its maximum size has been reached) always causes the Windows OS audit policy unit inability to trace (i.e. write) a generated audit event record. Furthermore, after the Windows OS machine is rebooted from the crash, only an administrator is allowed to logon.

To submit audit event records directly to the Windows OS security audit store service [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx), the Windows OS audit policy unit obtains a handle through the [ElfrRegisterEventSourceW()](http://msdn.microsoft.com/en-us/library/cc231428.aspx) interface of the Windows OS security audit store service by specifying the “Security” as the ModuleName parameter. The [ElfrRegisterEventSourceW()](http://msdn.microsoft.com/en-us/library/cc231428.aspx) interface grants a valid handle to only the local system and it only grants the handle once after the Windows OS security audit store service starts. The Windows OS audit policy unit (as part of the LSASS.exe Windows OS process) runs in the security context of the local system.

### Persistent security log file

Every audit event record expressed in the [EVENT\_RECORD](http://msdn.microsoft.com/en-us/library/aa363769(VS.85).aspx) structure is processed by the Windows OS security audit store service’s “[EventRecordCallback](http://msdn.microsoft.com/en-us/library/aa363743(VS.85).aspx)” function associated with the [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx). These audit event records have either

* the SecurityProviderGuid (“54849625-5478-4994-a5ba-3e3b0328c30d”);
* the [RTLostEventsGuid](http://msdn.microsoft.com/en-us/library/cc280265(VS.85).aspx) (“6A399AE0-4BC6-4DE9-870B-3657F8947E7E”)

as their ProviderId, which is a field in the corresponding [EventHeader](http://msdn.microsoft.com/en-us/library/aa363759(VS.85).aspx) of each of these records.

If the ProviderId of an event record indicates the [RTLostEventsGuid](http://msdn.microsoft.com/en-us/library/cc280265(VS.85).aspx), then the event record is replaced with an audit record for the [Event ID 1101](http://technet.microsoft.com/en-us/library/cc727052.aspx) (EVENT\_AUDIT\_EVENTS\_DROPPED) “Audit events have been dropped by the transport” indicating one of the following three reasons:

* “Events were lost because there were no free buffers. Flushing to disk could not catch up with incoming events”;
* “One or more buffers were dropped because a real time consumer could not catch up”;’
* “The real time backup file was corrupt due to improper shutdown”.

To persist the audit event records expressed in the [EVENT\_RECORD](http://msdn.microsoft.com/en-us/library/aa363769(VS.85).aspx) structure into the local file system, the Windows OS security audit store service’s “[EventRecordCallback](http://msdn.microsoft.com/en-us/library/aa363743(VS.85).aspx)” function associated with the [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx) writes their data to the security log file in the local file system. By default, the name of the security log file is “Security.evtx”. This default security log file resides locally in the “$SystemRoot$\System32\Winevt\Logs\” directory. The default security log file would have been opened by the Windows OS security audit store service when the Windows OS security audit store service was creating the [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx). During the file opening for the default security log file, the FILE\_SHARE\_WRITE is not specified. Therefore, another subject would not be able to open the default security log file for the write operations. The default security descriptor for the default security log file grants access to only:

* the Windows OS security audit store service (S-1-5-80-880578595-1860270145-482643319-2788375705-1540778122);
* the local system;
* an administrator.

### Audit records written by the Windows OS security audit store service

When the audit event records for the following two audit events arrive directly from the Windows OS audit policy unit, the Windows OS security audit store service writes them directly to the default security log file.

* Event ID 521 (SE\_AUDITID\_UNABLE\_TO\_LOG\_EVENTS) “Unable to log events to security log:” indicating the failure problem, in terms of the status code, and the number of failed audits;
* Event ID 525 (SE\_AUDITID\_CRASHONAUDITFAIL\_RECOVERY) “Administrator recovered system from CrashOnAuditFail. LSA will now accept non-administrative logons. Some auditable activity might not have been recorded.”

When the Windows OS security audit store service shutdowns, it writes an audit record for the [Event ID 1100](http://technet.microsoft.com/en-us/library/cc727131.aspx) (EVENT\_SHUTDOWN) “The event logging service has shut down” directly to the default security log file.

When the default security log file is full (i.e. reaching its maximum size) and cannot be written except for one last audit record, the Windows OS security audit store service writes an audit record for the [Event ID 1104](http://technet.microsoft.com/en-us/library/cc774916.aspx) (EVENT\_AUDIT\_LOG\_FULL) “The security log is now full” directly to the default security log file.

When the default security log file is full (i.e. reaching its maximum size) and the administrator specified “Backup log automatically when full” policy and “retain old events” policy are set, the Windows OS security audit store service writes an audit record for the [Event ID 1105](http://technet.microsoft.com/en-us/library/cc727199.aspx) (EVENT\_AUTO\_BACKUP) “Event log automatic backup” directly to the default security log file, indicating

* that the default security log file is full (i.e. reaching its maximum size);
* the name of the automatically backup log file.

By default, the administrator specified “Backup log automatically when full” policy and “retain old events” policy are not set.

When the size of the default security log file exceeds the administrator specified non-zero warning level, the Windows OS security audit store service writes an audit record for the [Event ID 1103](http://technet.microsoft.com/en-us/library/cc774990.aspx) (EVENT\_AUDIT\_LOG\_EXCEEDS\_WARNING\_LEVEL) “The security log is now <the warning level> percent full” directly to the default security log file. By default, the administrator specified warning level value is zero.

When the default security log is cleared by an authorized subject, the Windows OS security audit store service writes an audit record for the [Event ID 1102](http://technet.microsoft.com/en-us/library/cc774973.aspx) (EVENT\_AUDIT\_LOG\_CLEARED) “The audit log was cleared” directly to the default security log file, indicating the following attributes of the authorized subject:

* Subject User SID;
* Subject User Name;
* Subject Domain Name;
* Subject Logon ID.

### Prevention of previously generated audit record modification

In the above, we have shown the lifecycle of an audit event record. In its start, the audit event record is generated by a responsible audit generating module possessing the [SeAuditPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK). In its end, the audit event record is written to the local security log file for storage by the Windows OS security audit store service. As the audit event record flows through the following elements:

* the Windows OS audit policy unit residing within the LSASS.exe Windows OS process;
* the kernel mode [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx);
* the “Etw Logger” Windows OS system thread for the “EventLog-Security” logger;
* the persistent “EtwRTEventLog-Security.etl” file, which is used for saving real-time buffered audit event records that could not be delivered to the authorized consumers for any reason;
* the [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx) of the Windows OS security audit store service (wevtsvc.dll);
* the “Security.evtx” default security log file and its automatically backup log file,

the above has shown that there is no Windows OS interface without a proper security check for an unauthorized subject to attempt to modify the audit event record during its lifecycle.

Furthermore, in the case where the [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx) is deployed, the audit event records are also transported to the ACS collector and then to the ACS database for centralized storage and analysis, as explained in the justification text for addressing the Commercial Grade OS Requirement Set “8.1.1.3” requirement. The ACS transportation to the ACS collector is natively [protected with the encryption provided by the Windows OS Kerberos security provider](http://technet.microsoft.com/en-us/library/bb309575.aspx). Therefore, modification during the ACS transportation is prevented.

Finally, in the off-line situation, an attempt to modify the content of the “Security.evtx” default security log file and its automatically backup log file is prevented due to the availability of the Windows OS BitLocker™ components for supporting the Full volume encryption.

As a result, this Commercial Grade OS Requirement Set “8.2.1.1” requirement is met.

## Addressing 8.2.1.2 “The OS shall provide the capability for authorized administrators to specify the specific actions to be taken upon audit storage exhaustion”

The Commercial Grade OS Requirement Set suggests the following set of actions for an administrator to select:

1. stop performing operations that are being audited;
2. overwrite oldest audit data;
3. automatically increase audit storage space;
4. automatically archive audit data;
5. disable auditing and continue to operate.

This requirement is addressed by the Windows OS as follows.

The maximum size of the local security log file can be configured by an administrator. It is defined by the “MaxSize” registry key value under the following registry key.

* HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security.

By default, when the security log file is full (i.e. reaching its maximum size), the Windows OS security audit store service overwrites the oldest audit event records with the latest audit event records in its internal “File Overwrite Oldest Chunk” function.

Therefore, by default, the Windows OS takes the “overwrite oldest audit data” action (i.e. action 2)) upon audit storage exhaustion.

An administrator can configure the “retain old events” policy through the “Retention” registry key value under the following registry key.

* HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security.

If the “retain old events” policy is set, then the latest audit event records are not written to the security log file when the security log file is full (i.e. reaching its maximum size). By default, the “retain old events” policy is not set.

An administrator can configure the “Backup log automatically when full” policy through the “AutoBackupLogFiles” registry key value under the following registry key.

* HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security.

If the “Backup log automatically when full” policy is set and the “retain old events” policy is set, then the content of the security log file is automatically copied to its automatically backup log file before the security log file is clear for reclaiming the available space again to write the latest audit event records, when the security log file is full (i.e. reaching its maximum size). If the “retain old events” policy is not set, then the “Backup log automatically when full” policy does not take any effect. By default, the “Backup log automatically when full” policy is not set.

In the case where the “Backup log automatically when full” policy is not set and the “retain old events” policy is set, the Windows OS effectively takes the “disable auditing and continue to operate” action (i.e. action 5)) upon audit storage exhaustion.

In the case where the “Backup log automatically when full” policy is set and the “retain old events” policy is set, the Windows OS effectively takes the “automatically archive audit data” action (i.e. action 4)) upon audit storage exhaustion.

An administrator can configure the “CrashOnAuditFail” policy through the “CrashOnAuditFail” registry key value under the following registry key.

* HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa.

If the “CrashOnAuditFail” policy is set, then the Windows OS audit policy unit crashes the Windows OS machine to produce [STATUS\_AUDIT\_FAILED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) (0xC0000244L) on the blue screen when it is unable to trace (i.e. write) a generated audit event record through the [NtTraceEvent()](http://msdn.microsoft.com/en-us/library/aa364137.aspx) interface of the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) with the “ETW\_NT\_FLAGS\_TRACE\_SECURITY” flag. A full security log file (i.e. its maximum size has been reached) always causes the Windows OS audit policy unit inability to trace (i.e. write) a generated audit event record. Furthermore, after the Windows OS machine is rebooted from the crash, only an administrator is allowed to logon.

In the case where the “CrashOnAuditFail” policy is set, the Windows OS effectively takes the “stop performing operations that are being audited” action (i.e. action 1)) upon audit storage exhaustion.

As a result, the Windows OS provides the capability for authorized administrators to specify the following specific actions to be taken upon audit storage exhaustion:

1. stop performing operations that are being audited;
2. overwrite oldest audit data;
3. <no action specified in this item>;
4. automatically archive audit data;
5. disable auditing and continue to operate.

Therefore, this Commercial Grade OS Requirement Set “8.2.1.2” requirement is met.

# Meeting the “Auditing Audit Storage Management Requirements”

In the Commercial Grade OS Requirement Set, there are 3 individual management requirements under the heading of “Auditing Audit Storage Management Requirements”. They are listed as “8.2.2.n”, where n = 1, 2, and 3.

## Addressing 8.2.2.1 “The OS shall provide an authorized administrator with the capability to specify actions to be taken upon audit storage exhaustion”

This requirement is addressed by the Windows OS as follows.

This requirement is addressed by the descriptions given in the justification text for addressing the Commercial Grade OS Requirement Set “8.2.1.2” requirement.

## Addressing 8.2.2.2 “The OS shall provide an authorized administrator with the capability to sort, select and review collected audit records based on identity and any audit information items specified in “8.1.1.4””

This requirement is addressed by the Windows OS as follows.

The Windows OS security audit store service provides two interfaces:

* [ElfrReadELW()](http://msdn.microsoft.com/en-us/library/cc231426.aspx);
* [EvtRpcRegisterLogQuery()](http://msdn.microsoft.com/en-us/library/cc205033.aspx),

to query the content of the security log file. Both interfaces require the caller subject to possess the EVT\_READ\_ACCESS access right in the security descriptor associated with the [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx) of the Windows OS security audit store service. By default, an administrator or a member of the “event log readers” group (S-1-5-32-573) has the EVT\_READ\_ACCESS access right. The security descriptor associated with the [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx) can be changed by an administrator using the “CustomSD” registry key value under the following registry key.

* HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security.

Having had the EVT\_READ\_ACCESS access right to the content of the security log file, the “event viewer” application (eventvwr.exe with elsext.dll or els.dll with elsext.dll) provides the authorized subject the ability

* to sort audit event records in the security log based on one of the following audit event record information items:
  + Date and Time;
  + Event ID;
  + Task Subcategory;
  + User;
  + Event Level;
  + Operational Code;
  + Process ID;
  + Thread ID;
  + Session ID;
  + Correlation ID;
  + Relative Correlation ID;
  + Event Source Name;
  + Keywords of Audit Success or Audit Failure;
* to filter audit event records in the security log based on one or more of the following audit event record information items:
  + Custom range of dates and times;
  + Event Level;
  + Event ID or Event ID range inclusion;
  + Event ID or Event ID range exclusion;
  + Keywords of Audit Success or Audit Failure;
  + User;
* to find a specific free form text string in the security log audit event records;
* to review the security log audit event records sequentially;
* to select an audit event record of the security log currently being reviewed.

As a result, this Commercial Grade OS Requirement Set “8.2.2.2” requirement is met.

## Addressing 8.2.2.3 “The OS shall provide an authorized administrator with the capability to archive audit data”

This requirement is addressed by the Windows OS as follows.

The Windows OS security audit store service provides two interfaces:

* [ElfrClearELFW()](http://msdn.microsoft.com/en-us/library/cc231416.aspx);
* [EvtRpcClearLog()](http://msdn.microsoft.com/en-us/library/cc205012.aspx),

to clear the content of the security log file, and, optionally, to back up the security log file content to the specified name of the backup file before the clear operation takes place. Both interfaces require the caller subject to possess the EVT\_CLEAR\_ACCESS access right in the security descriptor associated with the [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx) of the Windows OS security audit store service. By default, an administrator has the EVT\_CLEAR\_ACCESS access right. The security descriptor associated with the [“Security” channel](http://msdn.microsoft.com/en-us/library/aa385225(VS.85).aspx) can be changed by an administrator using the “CustomSD” registry key value under the following registry key.

* HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Services\Eventlog\Security.

As a result, this Commercial Grade OS Requirement Set “8.2.2.3” requirement is met.

# Meeting the “Auditing Audit Storage Audit Requirements”

In the Commercial Grade OS Requirement Set, there is 1 individual audit requirement under the heading of “Auditing Audit Storage Audit Requirements”. It is listed as “8.2.3.1”.

## Addressing 8.2.3.1 “The OS shall provide the ability to audit the deleting or archiving of audit data”

This requirement is addressed by the Windows OS as follows.

The Windows OS security audit store service provides two interfaces:

* [ElfrClearELFW()](http://msdn.microsoft.com/en-us/library/cc231416.aspx);
* [\_EvtRpcClearLog()](http://msdn.microsoft.com/en-us/library/cc205012.aspx),

to clear the content of the security log file, and, optionally, to back up the security log file content to the specified name of the backup file before the clear operation takes place. If either interface handles its request successfully, then the Windows OS security audit store service writes an audit record for the [Event ID 1102](http://technet.microsoft.com/en-us/library/cc774973.aspx) (EVENT\_AUDIT\_LOG\_CLEARED) “The audit log was cleared” directly to the default security log file, indicating the following attributes of the authorized subject who has attempted the interface:

* Subject User SID;
* Subject User Name;
* Subject Domain Name;
* Subject Logon ID.

As a result, this Commercial Grade OS Requirement Set “8.2.3.1” requirement is met.

# Appendix A: Administrative Privileged Application Launching Service

When a subject wishes to launch a named application which resides on the Windows OS local file system or a Windows OS remote share across the network, the subject requests the Windows OS Administrative Privileged Application Launching Service to determine its “appropriateness” from a security relevant perspective before letting the subject launch the application in a security context that “matches” the intents of the application according to certain predefined security policies. The security relevant perspective for this service focuses on the interactive user’s runtime protection from launching possibly unintended applications accidentally. As a by-product, malicious software (pretending as an application) is less likely to be in a position to use an interactive logon user’s security context to launch privileged system applications without the user’s knowledge.

The determination of the Administrative Privileged Application Launching Service depends on a number of necessary reliable information items associated with the named application in a subject’s request to launch that application. These information items are listed as follows.

* Attributes of individual applications specified in the local trusted “named application attribute” database:
  + “RunAsAdmin”
    - The named application needs an administrator’s privileges to run correctly and it would not run correctly in a standard user account’s security context;
  + “RunAsHighest”
    - The named application runs in the most privileged security context available to the interactive user based on the user attributes granted to the user during his/her logon so that the named application runs with the standard user account’s privileges for a standard user account and the named application runs with the administrator’s privileges for an administrative user account after the administrator provides an explicit consent;
  + “RunAsInvoker”
    - The named application simply runs in the security context of its interactive invoker without elevation of any privileges which may have been granted to the user’s user account during his/her logon;
  + “NoSignatureCheck”
    - The PK-based (public key based) signature validation check is skipped for the named application so that the application is described as a unsigned executable in the elevation prompt which asks the consent from the interactive user;
  + “UIAccess”
    - The named application needs the UI Access right being made available to the process which will host the application to run in an elevated security context of the interactive user.
  + “GenericInstaller”
    - The named application belongs to a class of generically-identified software installers, which are meant to be used by an administrative user account only;
  + “SpecificInstaller”
    - The named application is a specific software installer, which are meant to be used by an administrative user account only, for certain software modules;
  + “SpecificNonInstaller”
    - The named application is not a software installer.

Typically, the associated attributes for an individual application are recommended by the creator of the application. However, it is the operating system product’s responsibility to protect the local trusted “named application attribute” database from a modification attempted by any untrusted subject. The word “database” is used within this paper in a generic sense. An operating system product may choose different and multiple mechanisms to implement the “named application attribute” database which has to be trusted. For the Windows OS, the local trusted “named application attribute” database comes from two sources:

* the Application Compatibility Database (eg a file with \*.sdb extension shipped with the Windows OS);
* the side-by-side assembly manifests for the corresponding named applications.

In the case of Application Compatibility Database \*.sdb files, they are created centrally by Microsoft as part of the shipped product building process. In a running system after the proper installation of Windows OS, they reside in the “$SystemRoot$\AppPatch” directory of the local file system. These files have the default file object security descriptor, and the directory has the default file directory object security descriptor as follows.

* The default file object security descriptor has the Windows OS Trusted Installer (S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464) as the owner. It grants the built-in “Administrators” group, the local system, and the built-in “Users” group only the generic read and execute access rights.
* The default file directory object security descriptor has the Windows OS Trusted Installer (S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464) as the owner. It grants the built-in “Administrators” group and the local system only the modify access rights, and it grants the built-in “Users” group only the generic read and execute access rights.

In the case of side-by-side assembly manifests, they are created by individual developers/publishers of the corresponding applications. In a running system after the proper installation of the Windows OS, they reside in the “$SystemRoot$\winsxs\Manifests” directory of the local file system if they are not embedded in the corresponding binary executable files. The manifests and the directory are similarly protected, with the default file object and file directory object security descriptors, from unauthorized modifications. Additionally, manifests are digitally signed as described in <http://msdn2.microsoft.com/en-us/library/acz3y3te(VS.80).aspx>.

The Administrative Privileged Application Launching Service also needs to be aware of the current security context that the requesting subject is running in when making the determination to let the subject launch the requested application. The possible security contexts that the service would detect independently for the requesting subject are described as follows.

* The requesting subject is already running in an elevated security context of the user account identity currently associated with the requesting subject with the granted administrative privileges[[32]](#footnote-33).
* The requesting subject is an administrator, but not running in an elevated security context with any administrative privileges;
* The requesting subject is a user account possessing some administrative privileges, but not running in an elevated security context with any of these administrative privileges[[33]](#footnote-34);
* The requesting subject is a standard user, not possessing any administrative privileges.

Additionally, the Administrative Privileged Application Launching Service also supports a number of security policies which may be configured by only an administrator:

* the “[Administrator Approval Mode](http://msdn.microsoft.com/en-us/library/cc232765.aspx)” policy
  + if enabled (which is the default), this policy says
    - all user accounts execute their locally created Windows OS processes with least privilege, including administrators, so that a user interactively logged onto a user account is required to provide consent to elevate privileged operations (such as software installations) that require administrator privileges;
* the “Safe Locations of UI Access Applications” policy
  + if enabled (which is the default), this policy says
    - applications, that need the UI Access right being made available to a Windows OS process which will host the applications to run in an elevated security context of the interactive user, must reside in a known safe location within the local file system where standard user (i.e. untrusted) subjects do not have the write or modify access;
* the “[Elevation Prompt for Administrators](http://msdn.microsoft.com/en-us/library/cc232761.aspx)” policy
  + this policy supports the following “prompt type” values:
    - “Prompt for consent” (which is the default), where a local operation, that requires elevation of privilege, causes the prompting for the consenting administrator to select either “Permit” or “Deny”;
    - “Prompt for credentials”, where a local operation, that requires elevation of privilege, causes the prompting for the consenting user to enter an administrator’s user name and password;
    - “Elevate without prompting”, where the interactively logged on administrator simply performs any local operation, which requires elevation, without consenting or entering credentials;
* the “[Elevation Prompt for Standard Users](http://msdn.microsoft.com/en-us/library/cc232762.aspx)” policy
  + this policy supports the following “prompt type” values:
    - “Prompt for credentials” (which is the default for a “Home” SKU (i.e. Windows Vista Home Premium or Windows Vista Home Basic)), where a local operation, that requires elevation of privilege, causes the prompting for the consenting standard user to enter an administrator’s user name and password;
    - “Automatically deny elevation requests” (which is the default for an “Enterprise” SKU (i.e. Windows Vista Business or Windows Vista Ultimate)), where an access denied error message is returned to the interactively logged on standard user when a local operation, that requires elevation of privilege, occurs;
* the “[Prompting on the Secure Display Area](http://msdn.microsoft.com/en-us/library/cc206334.aspx)” policy
  + if enabled (which is the default), this policy says
    - when prompting under the “Elevation Prompt for Administrators” policy or the “Elevation Prompt for Standard Users” policy, the prompting must occur on the secure display area;
* the “[Validate PK-based (public key based) Signature for Privileged Applications](http://msdn.microsoft.com/en-us/library/cc232764.aspx)” policy
  + if enabled (which is not the default), this policy says
    - an interactive application that requests elevation of privilege must have a valid public key signature to show that the application is the same copy as the one, which was created and signed by its publisher, before it is started;
* the “[Application Installation Detection](http://msdn.microsoft.com/en-us/library/cc206330.aspx)” policy
  + if enabled, this policy says
    - application installation packages that require an elevation of privilege to install locally are heuristically detected for triggering the elevation prompting according to the “Elevation Prompt for Administrators” policy or the “Elevation Prompt for Standard Users” policy

(by default, this policy is not enabled for an “Enterprise” SKU (i.e. Windows Vista Business or Windows Vista Ultimate) because the Group Policy Software Install (GPSI) is configured by the centralized administrator so that the involvement of the interactive user decision is not necessary)

(by default, this policy is enabled for a “Home” SKU (i.e. Windows Vista Home Premium or Windows Vista Home Basic));

* the “Secure Credential Prompting” policy
  + if enabled (which is not the default), this policy says
    - the entering of a user account’s credentials occurs on the secure display area through the use of the trusted path hot key (e.g. CTRL + ALT + DELETE) based on a permitted request to the window logon state maintaining service instance to request the window logon user interface service to prompt a dialog box on the secure display area, where the dialog box asks the interactive logged on user to supply his/her credential information (such as the user name with a password or the smart card PIN) corresponding to a credential provider installed on the local machine, to prevent a Trojan horse or other types of malicious code from stealing the credentials[[34]](#footnote-35);
* the “Allow security contexts with UI access right other than an administrator’s to override the “Prompting on the Secure Display Area” behavior” policy
  + if enabled (which is not the default), this policy says
    - a security context, other than an administrator’s, having been granted the UI access right may request to override the “Prompting on the Secure Display Area” behavior (in the window terminal session where the requesting subject resides in).

These polices would influence the security behaviors of the Administrative Privileged Application Launching Service and therefore provide additional variations of the service’s security functionalities.

The Administrative Privileged Application Launching Service also provides the exclusive service to

* the Windows OS Distributed Component Object Activation and Launching Service (rpcss.dll);
* the Windows OS Software Installer Service (msiexec.exe),

for configuring a security context, belong to a requesting subject, to

* activate a named distributed software library component object in the window terminal session where the subject resides in;
* perform an install or uninstall action for the specified software package or software patch updates on the local machine,

respectively.

In the [Microsoft publication: “Security Functional Assertions of the “User Interaction based on Windowing” Scenario of a Modern Operating System”](http://download.microsoft.com/download/e/d/b/edbb17fb-580d-49a4-b66c-8726cf446a86/User%20Interaction.doc), the security functionality variations of the Administrative Privileged Application Launching Service are enumerated based on:

* the associations with the logical local trusted “named application attribute” database;
* the associations with the security policies;
* the named application “RunAsAdmin” and “NoSignatureCheck” attribute values;
* the named application “RunAsAdmin” and “UIAccess” attribute values;
* the named application “RunAsAdmin” attribute value;
* the named application “RunAsHighest” and “NoSignatureCheck” attribute values;
* the named application “RunAsHighest” and “UIAccess” attribute values;
* the named application “RunAsHighest” attribute value;
* the named application “RunAsInvoker” and “NoSignatureCheck” attribute values;
* the named application “RunAsInvoker” and “UIAccess” attribute values;
* the named application “RunAsInvoker” attribute value;
* the named application “GenericInstaller” or “SpecificInstaller” attribute value;
* the named application “SpecificNonInstaller” attribute value;
* the non-existence of a named application attribute value;
* the case where the “Administrator Approval Mode” policy is disabled;
* which specific display area where a consent dialog is displayed to the interactive user;
* the command launching information database[[35]](#footnote-36);
* the distributed software library component object activation managed by the Distributed Component Object Activation and Launching Service;
* the requests that the named distributed software library component object, mediated by the Distributed Component Object Activation and Launching Service, is treated as “RunAsAdmin”;
* the requests that the named distributed software library component object, mediated by the Distributed Component Object Activation and Launching Service, is treated as “RunAsHighest”;
* the requests that the named distributed software library component object, mediated by the Distributed Component Object Activation and Launching Service, is treated as “RunAsInvoker”;
* the authorization check of software installation managed by the Software Installer Service;
* the requests that the software package or patch update install/uninstall action, mediated by the Software Installer Service, is treated as “RunAsAdmin”;
* the requests that the software package or patch update install/uninstall action, mediated by the Software Installer Service, is treated as “RunAsHighest”;
* the requests that the software package or patch update install/uninstall action, mediated by the Software Installer Service, is treated as “RunAsInvoker”.

The following lists the possible consent dialogs that are displayed to the interactive user. The precise conditions under which these consent dialogs appear are stated in the [Microsoft publication: “Security Functional Assertions of the “User Interaction based on Windowing” Scenario of a Modern Operating System”](http://download.microsoft.com/download/e/d/b/edbb17fb-580d-49a4-b66c-8726cf446a86/User%20Interaction.doc).

* A consent dialog is displayed to the interactive user for requesting the credential of an administrative user account:
  + the consent dialog displaying to the interactive user indicates the following message texts:
    - “An unidentified program wants access to your computer”;
    - “Don't run the program unless you know where it's from or you've used it before”;
  + the consent dialog includes a “submit” button with the text string “I trust this program and I know where it's from or I've used it before”, for the user to acknowledge;
  + the consent dialog includes a “cancel” button with the text string “I don't know where this program is from or what it's for”, for the user to acknowledge;
  + either
    - in the case where the user selects the “submit” button and supplies the valid credential for the administrative user account,
      * the process, which will host the application, runs in an elevated security context of the administrative user account corresponding to the valid credential provided by the interactive user through the prompt;
    - in the case where the user selects the “submit” button and supplies an invalid credential for the administrative user account,
      * an error indicating that the supplied credential is not valid for logging onto the administrative user account;
    - in the case where the user selects the “cancel” button,
      * an error indicating that the user has canceled the request.
* A consent dialog is displayed to the interactive user:
  + the consent dialog displaying to the interactive user indicates the following message texts:
    - “An unidentified program wants access to your computer”;
    - “Don't run the program unless you know where it's from or you've used it before”;
  + the consent dialog includes a “submit” button with the text string “I trust this program and I know where it's from or I've used it before”, for the user to acknowledge;
  + the consent dialog includes a “cancel” button with the text string “I don't know where this program is from or what it's for”, for the user to acknowledge;
  + either
    - in the case where the user selects the “submit” button,
      * the process, which will host the application, runs in an elevated security context of the requesting subject;
    - in the case where the user selects the “cancel” button,
      * an error indicating that the user has canceled the request.
* A consent dialog is displayed to the interactive user for requesting the credential of an administrative user account:
  + the consent dialog displaying to the interactive user indicates one of the following sets of message texts:
    - “The operating system needs your permission to continue”, “If you started this action, continue”, “The operating system vendor's certificate information”;
    - “A program needs your permission to continue”, “If you started this program, continue”, “Unidentified Publisher’s certificate information”;
    - “An unidentified program wants access to your computer”, “Don't run the program unless you know where it's from or you've used it before”, “Unidentified Publisher’s certificate information”;
  + the consent dialog includes a “submit/continue” button for the user to acknowledge the execution of the application being informed in the dialog;
  + the consent dialog includes a “cancel” button for the user to acknowledge the cancelation of the execution of the application being informed in the dialog;
  + either
    - in the case where the user selects the “submit/continue” button and supplies the valid credential for the administrative user account,
      * the process, which will host the application, runs in an elevated security context of the administrative user account corresponding to the valid credential provided by the interactive user through the prompt;
      * the process, which will host the application, is additionally granted the high integrity level and the UI access right by the service, subject to the “Safe Locations of UI Access Applications” policy as follows:
        + the process is additionally granted the high integrity level and the UI access right in the case where the “Safe Locations of UI Access Applications” policy is disabled;
        + the process is additionally granted the high integrity level and the UI access right in the case where the “Safe Locations of UI Access Applications” policy is enabled and the binary file of the named application resides in a known safe location within the local file system where standard user (i.e. untrusted) subjects do not have the write or modify access;
        + the process is not additionally granted the high integrity level or the UI access right, otherwise;
      * the binary image file, from which the launched process has been created, is really the application binary file whose PK-based signature has been validated;
    - in the case where the user selects the “submit” button and supplies an invalid credential for the administrative user account,
      * an error indicating that the supplied credential is not valid for logging onto the administrative user account;
    - in the case where the user selects the “cancel” button,
      * an error indicating that the user has canceled the request.
* A consent dialog is displayed to the interactive user:
  + the consent dialog displaying to the interactive user indicates one of the following sets of message texts:
    - “The operating system needs your permission to continue”, “If you started this action, continue”, “The operating system vendor's certificate information”;
    - “A program needs your permission to continue”, “If you started this program, continue”, “Unidentified Publisher’s certificate information”;
    - “An unidentified program wants access to your computer”, “Don't run the program unless you know where it's from or you've used it before”, “Unidentified Publisher’s certificate information”;
  + the consent dialog includes a “submit/continue” button for the user to acknowledge the execution of the application being informed in the dialog;
  + the consent dialog includes a “cancel” button for the user to acknowledge the cancelation of the execution of the application being informed in the dialog;
  + either
    - in the case where the user selects the “submit/continue” button,
      * the process, which will host the application, runs in an elevated security context of the requesting subject;
      * the process, which will host the application, is additionally granted the high integrity level and the UI access right by the service, subject to the “Safe Locations of UI Access Applications” policy as follows:
        + the process is additionally granted the high integrity level and the UI access right in the case where the “Safe Locations of UI Access Applications” policy is disabled;
        + the process is additionally granted the high integrity level and the UI access right in the case where the “Safe Locations of UI Access Applications” policy is enabled and the binary file of the named application resides in a known safe location within the local file system where standard user (i.e. untrusted) subjects do not have the write or modify access;
        + the process is not additionally granted the high integrity level or the UI access right, otherwise;
      * the binary image file, from which the launched process has been created, is really the application binary file whose PK-based signature has been validated;
    - in the case where the user selects the “cancel” button,
      * an error indicating that the user has canceled the request.
* A consent dialog is displayed to the interactive user for requesting the credential of an administrative user account:
  + the consent dialog displaying to the interactive user indicates one of the following sets of message texts:
    - “The operating system needs your permission to continue”, “If you started this action, continue”, “The operating system vendor's certificate information”;
    - “A program needs your permission to continue”, “If you started this program, continue”, “Unidentified Publisher’s certificate information”;
    - “An unidentified program wants access to your computer”, “Don't run the program unless you know where it's from or you've used it before”, “Unidentified Publisher’s certificate information”;
  + the consent dialog includes a “submit/continue” button for the user to acknowledge the execution of the application being informed in the dialog;
  + the consent dialog includes a “cancel” button for the user to acknowledge the cancelation of the execution of the application being informed in the dialog;
  + either
    - in the case where the user selects the “submit/continue” button and supplies the valid credential for the administrative user account,
      * the process, which will host the application, runs in an elevated security context of the administrative user account corresponding to the valid credential provided by the interactive user through the prompt;
      * the binary image file, from which the launched process has been created, is really the application binary file whose PK-based signature has been validated;
    - in the case where the user selects the “submit” button and supplies an invalid credential for the administrative user account,
      * an error indicating that the supplied credential is not valid for logging onto the administrative user account;
    - in the case where the user selects the “cancel” button,
      * an error indicating that the user has canceled the request.
* A consent dialog is displayed to the interactive user:
  + the consent dialog displaying to the interactive user indicates one of the following sets of message texts:
    - “The operating system needs your permission to continue”, “If you started this action, continue”, “The operating system vendor's certificate information”;
    - “A program needs your permission to continue”, “If you started this program, continue”, “Unidentified Publisher’s certificate information”;
    - “An unidentified program wants access to your computer”, “Don't run the program unless you know where it's from or you've used it before”, “Unidentified Publisher’s certificate information”;
  + the consent dialog includes a “submit/continue” button for the user to acknowledge the execution of the application being informed in the dialog;
  + the consent dialog includes a “cancel” button for the user to acknowledge the cancelation of the execution of the application being informed in the dialog;
  + either
    - in the case where the user selects the “submit/continue” button,
      * the process, which will host the application, runs in an elevated security context of the requesting subject;
      * the binary image file, from which the launched process has been created, is really the application binary file whose PK-based signature has been validated;
    - in the case where the user selects the “cancel” button,
      * an error indicating that the user has canceled the request.
* A consent dialog is displayed to the interactive user:
  + the consent dialog includes the following message texts:
    - “A specific application program has been blocked”;
    - “Your administrator set policy to block this application program”;
    - “Untrusted Publisher’s certificate information”;
  + the consent dialog does not include a “submit” button, but only a “cancel” button for the user to acknowledge;
  + an error indicating that the user has canceled the request.

# Appendix B: Addressing Certain Interesting Security Functional Requirements

Since it publication, we received feedback on [Version 1.0 of this paper](http://download.microsoft.com/download/D/7/1/D7158253-CE22-4CB3-B622-E3460AB2B9B1/CommercialOSSecFunReqsPublic.docx) concerning additional security functional requirements. In this version (Version 2.0), we will address a selection of those requirements in the following areas, as additional to the Commercial Grade OS Requirement Set:

* Security Architecture Separation, Isolation, and Least Privilege;
* Security Architecture Distributed Architectures;
* Access Control Polices Discretionary Access Control Policy;
* Access Control Polices Mandatory Integrity Control Policy;
* Access Control Polices Mandatory Access Control Policy for User File/Directory Objects;
* Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication;
* Identification and Authentication User Interface Security;
* Information Protection Residual Information;
* Information Protection Resource Control;
* Information Protection Self Testing;
* Information Protection Persistent Storage Data Availability;
* Information Protection Maintenance Mode Access Control;
* Import/Export of Data Exported Data;
* Import/Export of Data Imported Data;
* Auditing Audit Collection;
* Auditing Audit Storage.

In this appendix, we present a comprehensive set of technical justifications to explain the manners in which Windows Vista and Windows Server 2008 collectively address these suggested security functional requirements.

# Meeting Additional “Security Architecture Separation, Isolation, and Least Privilege Functional Requirements”

The Commercial Grade OS Requirement Set already has 5 individual functional requirements under the heading of “Security Architecture Separation, Isolation, and Least Privilege Functional Requirements”. They are listed as “1.1.1.n”, where n = 1, 2, 3, 4, and 5.

There are 5 additional individual functional requirements under the heading of “Security Architecture Separation, Isolation, and Least Privilege Functional Requirements” in this appendix. They are listed as “1.1.1.n”, where n = 6, 7, 8, 9, and 10.

## Addressing 1.1.1.6 “The OS Memory protection shall provide specific capabilities”

The “1.1.1.6” requirement requires the following OS memory protection capabilities:

* “Protect addresses of storage units so that unauthorized users can not access protected storage units;
* Provide various types of protection to the operation of protected storage units. The basic protection types are “read / write” and “read-only”. If non- “read/write” storage units that are accessed for “read/write” operation, the system should raise alarm immediately or suspend the process;
* Logical isolation approach can be adopted for memory protection. Specific approaches include boundary address register protection approach, memory identification approach; lock protection approach and flag bit protection approach”.

The Windows OS also addresses the “1.1.1.6” requirement due to the mechanisms for enabling its “User mode kernel mode context switching” and “Process memory virtualization” levels of protection in its “defense in depth” concept. Please see the justification text for addressing the Commercial Grade OS Requirement Set “1.1.1.1” requirement for detail.

We note that the logical isolation approach adopted by the Windows OS is not any of those specified in the “1.1.1.6” requirement. The Windows OS’s approach includes the use of certain privileged processor architecture instructions. The processor architecture ensures that the use of these privileged instructions is limited to code (i.e. kernel mode code) which is executed only when the current privilege level of the processor architecture is at the highest. The user mode code is executed only when the current privilege level of the processor architecture is at the lowest. The isolation is accomplished by the “User mode kernel mode context switching”, which relies on the use of specific instructions of the underlying hardware processor architecture to let the processor architecture reset its current privilege level.

We believe that an explicit allowance of the possibility for a compliant OS product to implement its desired logical isolation approach for memory protection may be a better way to specify the “1.1.1.6” requirement.

## Addressing 1.1.1.7 “Separation of operating system programs (execution contexts) from user programs should be realized in the OS”

The “1.1.1.7” requirement additionally requires that “Virtual address space of a process should be at least divided into two sections: user space and system space, and their separation should be static. The operating system residing in the memory should be shared by all processes; and user processes should be separated; the operation of writing to system section when the process is under the user mode should be forbidden, while the operation of reading and writing to all virtual space when the process is under the system mode should be allowed”.

The “1.1.1.7” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “1.1.1.2” requirement “The OS shall ensure that all security policies are enforced before each security function is allowed to proceed”.

The Windows OS also addresses the “1.1.1.7” requirement due to the mechanisms for enabling its “User mode kernel mode context switching” and “Process memory virtualization” levels of protection in its “defense in depth” concept. Please see the justification text for addressing the Commercial Grade OS Requirement Set “1.1.1.2” requirement for detail.

## Addressing 1.1.1.8 “In a single-user system, the OS memory protection shall prevent user subject processes from affecting the OS”

The “1.1.1.8” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “1.1.1.3” requirement “The OS shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects”.

The Windows OS also addresses the “1.1.1.8” requirement because of the effectiveness of its “User mode kernel mode context switching” level of protection in its “defense in depth” concept. Please see the justification text for addressing the Commercial Grade OS Requirement Set “1.1.1.3” requirement for detail.

## Addressing 1.1.1.9 “In a multi-user system, the OS memory protection shall prevent the interference between user subjects within the OS”

The “1.1.1.9” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “1.1.1.4” requirement “The OS shall enforce separation between the security domains of subjects”.

The Windows OS also addresses the “1.1.1.9” requirement because of the effectiveness of its “User mode kernel mode context switching” and “Process memory virtualization” levels of protection in its “defense in depth” concept. Please see the justification text for addressing the Commercial Grade OS Requirement Set “1.1.1.4” requirement for detail.

## Addressing 1.1.1.10 “System “back door” shall not be designed in the OS”

The “1.1.1.10” requirement additionally clarifies that “No access points of any kind which violate or circumvent security rules should be designed in on the excuse of maintenance, support or operation necessity or unexplained in the documentation”.

The Windows OS does not meet the “1.1.1.10” requirement if the requirement is taken literally.

Specifically, the “1.1.1.10” requirement does not specify a security rule that allows an authorized administrator or user account operator to reset the user password of a user account (i.e. the “1.1.1.10” requirement does not explicitly allow a requirement similar to the Commercial Grade OS Requirement Set “3.1.2.5” requirement). Therefore, the ability of allowing an authorized administrator or a user account operator to reset the user password of a user account violates and circumvents the security rules of the “1.1.1.10” requirement, even though it is a desired capability that the IT helpdesks of many Windows OS customer organizations rely on for their maintenance, support and operation of user password management. It is also documented by Microsoft, for example, as shown in <http://technet.microsoft.com/en-us/library/cc732954.aspx>. However, if implemented appropriately, the capability of allowing an authorized administrator or a user account operator to reset the user password of a user account does not present a vulnerability to the compliant OS product.

We believe that the intent of the “1.1.1.10” requirement is best addressed as a security assurance requirement rather than as a security functional requirement (which is the current situation). For the specific security assurance requirements, we recommend the use of the Misuse analysis (AVA\_MSU) and Vulnerability analysis (AVA\_VLA) assurance requirements under the AVA: Vulnerability assessment assurance class of [ISO 15408 Part 3 “Information technology -- Security techniques -- Evaluation criteria for IT security -- Part 3: Security assurance requirements”](http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=40614). Alternatively, using the Vulnerability analysis (AVA\_VAN) assurance requirements under the AVA: Vulnerability assessment assurance class of the “[Common Criteria Part 3: Security assurance requirements](http://www.commoncriteriaportal.org/files/ccfiles/CCPART3V3.1R2.pdf)” should also be better suited for addressing the intent of the “1.1.1.10” requirement.

# Meeting Additional “Security Architecture Distributed Architectures Functional Requirements”

The Commercial Grade OS Requirement Set already has 5 individual functional requirements under the heading of “Security Architecture Distributed Architectures Functional Requirements”. They are listed as “1.2.1.n”, where n = 1, 2, 3, 4, and 5.

There is 1 additional individual functional requirement under the heading of “Security Architecture Distributed Architectures Functional Requirements” in this appendix. It is listed as “1.2.1.6”.

## Addressing 1.2.1.6 “The OS shall achieve the basic protection to security-relevant data transmission, data separation transmission and data integrity protection within the OS”

For the “basic protection to security-relevant data transmission” and “basic protection to security-relevant data separation transmission” aspects”, the “1.2.1.6” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “1.2.1.2” requirement “The OS shall protect security-relevant data from disclosure while being transmitted to a remote part of the OS through the use of OS provided cryptographic services”.

The Windows OS also addresses the “basic protection to security-relevant data transmission” and “basic protection to security-relevant data separation transmission” aspects of the “1.2.1.6” requirement.

For the “basic protection to security-relevant data integrity protection” aspect, the “1.2.1.6” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “1.2.1.3” requirement “The OS shall detect modification and insertion of security-relevant received from a remote part of the OS through the use of OS provided cryptographic services”.

The Windows OS also addresses the “basic protection to security-relevant data integrity protection” aspect of the “1.2.1.6” requirement.

# Meeting Additional “Security Architecture Distributed Architectures Management Requirements”

The Commercial Grade OS Requirement Set already has 1 individual management requirement under the heading of “Security Architecture Distributed Architectures Management Requirements”. It is listed as “1.2.2.1”

There are 7 additional individual management requirements under the heading of “Security Architecture Distributed Architectures Management Requirements” in this appendix. They are listed as “1.2.2.n”, where n = 2, 3, 4, 5, 6, 7, and 8.

## Addressing 1.2.2.2 “The OS shall achieve the protection to usability, confidentiality, and integrity of output security-relevant data”

The “1.2.2.2” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “1.2.2.1” requirement “The OS shall provide the ability for an authorized administrator to remotely manage the OS”.

The Windows OS also addresses the “1.2.2.2” requirement due to the Windows OS means for secure remote administration using the Windows OS provided cryptographic services for addressing the Commercial Grade OS Requirement Set “1.2.1.5” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “1.2.2.1” and “1.2.1.5” requirements for detail.

## Addressing 1.2.2.3 “The OS shall provide an installation mechanism for the setting and upgrade of configuration parameters”

The “1.2.2.3” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “1.2.2.1” requirement “The OS shall provide the ability for an authorized administrator to remotely manage the OS”.

The Windows OS also addresses the “1.2.2.3” requirement. [The group policy objects processed by the Windows OS group policy client service (gpsvc.dll)](http://www.microsoft.com/downloads/details.aspx?FamilyID=41dc179b-3328-4350-ade1-c0d9289f09ef&DisplayLang=en) include the “setting and upgrade of configuration parameters” aspects specified in the “1.2.2.3” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “1.2.2.1” requirement for detail.

## Addressing 1.2.2.4 “Before the initialization and the implementation of protection to security-relevant data structure, security policy attributes of users and administrators should be defined”

The “1.2.2.4” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “1.2.2.1” requirement “The OS shall provide the ability for an authorized administrator to remotely manage the OS”.

“The Windows OS also addresses the “1.2.2.4” requirement for a managed Windows OS machine within a Windows OS domain/forest or for a stand-alone unmanaged Windows OS machine.

A managed Windows OS machine within a Windows OS domain/forest automatically receives and processes centrally distributed system and security configuration policies through its Windows OS group policy client service (gpsvc.dll) after it successfully starts up, as explained in the “Group policy processing at a local computer” section of this paper. The Windows OS group policy client service processing occurs before any user is allowed a logging on attempt.

For a stand-alone unmanaged Windows OS machine, its administrator also has [the ability to define and configure local group policy objects](http://msdn.microsoft.com/en-us/library/bb530196.aspx). In the similar manner, the Windows OS group policy client service (gpsvc.dll) also processes these local group policy objects. This processing occurs after the start up of the Windows OS machine, but before any user is allowed a logging on attempt.

Please see the justification text for addressing the Commercial Grade OS Requirement Set “1.2.2.1” requirement for detail.

## Addressing 1.2.2.5 “After the completion of OS installation and before the access by ordinary users, the OS should configure responsibilities of initial users and administrators, root directory, audit parameters, system audit trail setup, and appropriate access control on files and directories”

The “1.2.2.5” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “1.2.2.1” requirement “The OS shall provide the ability for an authorized administrator to remotely manage the OS”.

“The Windows OS also addresses the “1.2.2.5” requirement. [The group policy objects processed by the Windows OS group policy client service (gpsvc.dll)](http://www.microsoft.com/downloads/details.aspx?FamilyID=41dc179b-3328-4350-ade1-c0d9289f09ef&DisplayLang=en) include the configuration aspects specified in the “1.2.2.5” requirement.

Please see the justification text for addressing the Commercial Grade OS Requirement Set “1.2.2.1” requirement for detail.

## Addressing 1.2.2.6 “Application programs provided by the OS should (by default) be limited to the effective use of the OS, and only administrators can modify or replace application programs provided by the OS”

The Windows OS addresses the “1.2.2.6” requirement. In the Windows OS, its system service processes are managed by the [Windows OS service control manager (services.exe)](http://msdn.microsoft.com/en-us/library/ms685150(VS.85).aspx). By default, an authenticated interactive user (who is not an administrator) is granted only the following access rights to a Windows OS system service:

* [SERVICE\_QUERY\_CONFIG](http://msdn.microsoft.com/en-us/library/ms685981(VS.85).aspx);
* [SERVICE\_QUERY\_STATUS](http://msdn.microsoft.com/en-us/library/ms685981(VS.85).aspx);
* [SERVICE\_ENUMERATE\_DEPENDENTS](http://msdn.microsoft.com/en-us/library/ms685981(VS.85).aspx);
* [SERVICE\_INTERROGATE](http://msdn.microsoft.com/en-us/library/ms685981(VS.85).aspx).

The “automatic start” configuration value of an individual system service is stored in the “Start” registry key value for the system service’s registry key under “HKLM\SYSTEM\CurrentControlSet\ Services”. The actual program responsible for an individual system service is identified in the “ImagePath” registry key value for the system service’s registry key. By default, standard users are granted only the following access rights to the system service’s registry key:

* [KEY\_QUERY\_VALUE](http://msdn.microsoft.com/en-us/library/ms724878.aspx);
* [KEY\_ENUMERATE\_SUB\_KEYS](http://msdn.microsoft.com/en-us/library/ms724878.aspx);
* [KEY\_NOTIFY](http://msdn.microsoft.com/en-us/library/ms724878.aspx).

In a running system, after the proper installation of Windows OS, the Windows OS system executable image files (such as those of system services) reside in the “$SystemRoot$\System32” directory of the local file system. These files have the default file object security descriptor, and the directory has the default file directory object security descriptor as follows.

* The default file object security descriptor has the Windows OS Trusted Installer (S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464) as the owner. It grants the built-in “Administrators” group, the local system, and the built-in “Users” group only the generic read and execute access rights.
* The default file directory object security descriptor has the Windows OS Trusted Installer (S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464) as the owner. It grants the built-in “Administrators” group and the local system only the modify access rights, and it grants the built-in “Users” group only the generic read and execute access rights.

## Addressing 1.2.2.7 “The OS shall provide a mechanism for security administrator of operating system to generate detailed report on security parameters”

The Windows OS addresses the “1.2.2.7” requirement. The Windows OS [Resultant Set of Policy (RSoP)](http://technet.microsoft.com/en-us/library/cc778752.aspx) is an additional capability to the Windows OS Group Policy. The Windows OS RSoP is a query engine that polls existing policies and planned policies for an administrator. It also reports the results of those queries to the administrator. The RSoP reports include details about all policy settings that have been configured for a target Windows OS machine by an administrator.

## Addressing 1.2.2.8 “The OS environment shall be able to control the use of the OS control panel”

The Windows OS addresses the “1.2.2.8” requirement. Administrative tasks available in the Windows OS “Control Panel” are subject to the control of the Windows OS Administrative Privileged Application Launching Service, which is described in the “Appendix A: Administrative Privileged Application Launching Service” section of this paper.

Because of the default “[Administrator Approval Mode](http://msdn.microsoft.com/en-us/library/cc232765.aspx)” policy, all user accounts execute their locally created Windows OS processes with least privilege, including administrators, so that a user interactively logged onto a user account is required to provide consent to elevate privileged operations (such as the administrative tasks in the Windows OS “Control Panel”) that require administrator privileges.

The default “prompt type” of the “[Elevation Prompt for Administrators](http://msdn.microsoft.com/en-us/library/cc232761.aspx)” policy is “Prompt for consent”, where a local operation, that requires elevation of privilege, causes the prompting for the consenting administrator to select either “Permit” or “Deny”.

In a “Home” SKU (i.e. Windows Vista Home Premium or Windows Vista Home Basic), the default “prompt type” of the “[Elevation Prompt for Standard Users](http://msdn.microsoft.com/en-us/library/cc232762.aspx)” policy is “Prompt for credentials”, where a local operation, that requires elevation of privilege, causes the prompting for the consenting standard user to enter an administrator’s user name and password.

In an “Enterprise” SKU (i.e. Windows Vista Business or Windows Vista Ultimate), the default “prompt type” of the “[Elevation Prompt for Standard Users](http://msdn.microsoft.com/en-us/library/cc232762.aspx)” policy is “Automatically deny elevation requests”, where an access denied error message is returned to the interactively logged on standard user when a local operation, that requires elevation of privilege, occurs.

Examples of administrative tasks in the Windows OS “Control Panel” that require elevation of privilege are:

* Individual tasks under “Control Panel\Administrative Tools”
* “Set program access and computer defaults” under “Control Panel\Default Programs”;
* “Turn Windows features on or off” under “Control Panel\Programs and Features”;
* “Transfer files and settings” under “Control Panel\Welcome Center”;
* “Control Panel\Parental Controls”;
* “Control Panel\Add Hardware”;
* “Device Manager” under “Control Panel\System”;
* “Install/uninstall languages…” under “Control Panel\Regional and Language Options”;
* “Windows Firewall Settings” under “Control Panel\Security Center”;
* “Adjust the appearance and performance of Windows” under “Control Panel\Performance Information and Tools\Advanced Tools”;
* “Remote settings” under “Control Panel\System”;
* “Computer name, domain, and workgroup settings” under “Control Panel\System”;
* “System protection” under “Control Panel\System”;
* “Advanced system settings” under “Control Panel\System”;
* “Repair Windows using System Restore” under “Control Panel\Backup and Restore Center”;
* “Turn User Account Control on or off” under “Control Panel\User Accounts”;
* “Manage User Accounts” under “Control Panel\User Accounts”;
* “BitLocker Drive Encryption” under “Control Panel”;
* “Adjust font size (DPI)” under “Control Panel\Personalization”.

We note that the task launching control of the above administrative tasks is not the only access control for a user attempting to make security relevant management/configuration changes in the Windows OS. There are the corresponding Windows OS services which also have to meet their corresponding management requirements when they are handling a subject’s request for making security relevant configuration changes under their security policies. These Windows OS services are the specific resource managers for managing their corresponding resources. Their management/configuration interfaces are necessary non-bypassable. On the other hand, administrative tasks available in the system control panel can be bypassed because a subject may attempt the management/configuration interfaces of the Windows OS resource managers directly. The administrative tasks available in the system control actually also use the management/configuration interfaces of the Windows OS resource managers.

Specific Windows OS services are described and explained throughout this paper for their security relevant aspects.

# Meeting Additional “Security Architecture Distributed Architectures Audit Requirements”

The Commercial Grade OS Requirement Set already has 1 individual audit requirement under the heading of “Security Architecture Distributed Architectures Audit Requirements”. It is listed as “1.2.3.1”

There is 1 additional individual audit requirement under the heading of “Security Architecture Distributed Architectures Audit Requirements” in this appendix. It is listed as “1.2.3.2”.

## Addressing 1.2.3.2 “The OS environment shall be able to audit the use of system control panel”

The Windows OS addresses the “1.2.3.2” requirement. Typically, for an administrative task available in the Windows OS “Control Panel” mentioned in the justification text for addressing the “1.2.2.8” requirement, there is an application executable file residing in the “%windir%\system32” that need to be launched in the security context of the launching user subject. For example, the “%windir%\system32\SystemPropertiesComputerName.exe” application executable file corresponds to the “Computer name, domain, and workgroup settings” task under “Control Panel\System”. Actually, the “%windir%\system32\mmc.exe” [Microsoft Management Console (MMC)](http://msdn.microsoft.com/en-us/library/ms692748(VS.85).aspx) application with a [specific MMC snap-in (\*.msc) files](http://support.microsoft.com/kb/230263) is the most common way to launch an administrative task.

When an application executable file is launched in the security context of the launching user, the Windows OS process manager needs to create a Windows OS process object to house the application. As the Windows OS process manager completes the initialization of the Windows OS process for the application, it generates the [Event ID 4688](http://support.microsoft.com/kb/947226) (SE\_AUDITID\_ETW\_PROCESS\_ CREATED\_ value) “A new process has been created” audit record in the hard audit store to indicate the identity of the user subject launching the application and the name of the application.

We also note that as the corresponding Windows OS services handle a subject’s request for making security relevant management/configuration changes under their security policies, these Windows OS services also generate security relevant audit records where appropriate. The audit records that these Windows OS services generate are described and explained throughout this paper.

# Meeting Additional “Access Control Polices Discretionary Access Control Policy Management Requirements”

The Commercial Grade OS Requirement Set already has 3 individual management requirements under the heading of “Access Control Polices Discretionary Access Control Policy Management Requirements”. They are listed as “2.1.2.n”, where n = 1, 2, and 3.

There are 2 additional individual management requirements under the heading of “Access Control Polices Discretionary Access Control Policy Management Requirements” in this appendix. They are listed as “2.1.2.n”, where n = 4, and 5.

## Addressing 2.1.2.4 “Owner of an object shall be the only subject with the right to modify the access right to the object”

The “2.1.2.4” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “2.1.2.1” requirement “The OS shall allow only authorized administrators, object owners, and users with the DAC change authorization the ability to change the access permission associated with a named object”.

In its default behavior, the Windows OS allows object owners the ability to change the access permission associated with a named object. In reality, there is no effective countermeasure to prevent a determined authorized administrator to modify the access right to any object. In the Windows OS, [WRITE\_DAC](http://msdn.microsoft.com/en-us/library/aa379607.aspx) is one of the standard access control elements applicable to every named object. Under certain situations (e.g. for supporting recoverability), Windows OS resource managers grant WRITE\_DAC to authorized administrators by introducing WRITE\_DAC into the access mask for inclusion in an ACE, which identifies the “Administrators” local group as the trustee. Please see the justification text for addressing the Commercial Grade OS Requirement Set “2.1.2.1” requirement for detail.

Therefore, having the owner of the object as the only subject with the right to modify the access right to the object may not be realistic or feasible in certain Windows OS deployment scenarios. We recommend that “the OS shall allow only authorized administrators, object owners, and users with the DAC change authorization the ability to change the access permission associated with a named object”, which is the Commercial Grade OS Requirement Set “2.1.2.1” requirement”.

## Addressing 2.1.2.5 “Distribution of the control rights on an object to other subjects by the object’s owner shall be forbidden”

The “2.1.2.5” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “2.1.2.1” requirement “The OS shall allow only authorized administrators, object owners, and users with the DAC change authorization the ability to change the access permission associated with a named object”.

From our users’ feedback, some Windows OS users actually desire to limit the object creator/owner ability to modify access rights for specific objects. In the Windows OS, if a Windows OS resource manager does not wish to grant WRITE\_DAC to the object creator/owner subject of its object, then it specifies an explicit “creator owner rights” ACE in the DACL of the object, where the “creator owner rights” ACE does not include WRITE\_DAC in its access mask.

In the Windows OS, the support of delegation is critical. Some Windows OS users desire that granting the ability to modify the access rights to another identity needs to be available in the Windows OS. The [WRITE\_DAC](http://msdn.microsoft.com/en-us/library/aa379607.aspx) is just an access right similar to other access rights such as “read”, “write”, or “execute”, which may be granted or denied to a user account or user group. The access determination algorithm treats them in the same manner when determining a requester’s desired access to a named object.

We recommend that “the OS shall allow only authorized administrators, object owners, and users with the DAC change authorization the ability to change the access permission associated with a named object”, which is the Commercial Grade OS Requirement Set “2.1.2.1” requirement”.

# Meeting Additional “Access Control Polices Mandatory Integrity Control Policy Functional Requirements”

The Commercial Grade OS Requirement Set already has 6 individual functional requirements under the heading of “Access Control Polices Mandatory Integrity Control Policy Functional Requirements”. They are listed as “2.2.1.n”, where n = 1, 2, 3, 4, 5 and 6.

There are 3 additional individual functional requirements under the heading of “Access Control Polices Mandatory Integrity Control Policy Functional Requirements” in this appendix. They are listed as “2.2.1.n”, where n = 7, 8, and 9.

## Addressing 2.2.1.7 “Function on the protection of integrity of user data transmitted within the OS, such as inter-process communication data shall be provided: Integrity label shall be moved with data”

Due to the Commercial Grade OS Requirement Set “1.2.1.3” and “1.2.1.4” requirements, the Windows OS also addresses the first part of the “2.2.1.7” requirement, namely “Function on the protection of integrity of user data transmitted within the OS, such as inter-process communication data shall be provided”.

The Windows OS does not meet the second part of the “2.2.1.7” requirement, namely “Integrity label shall be moved with data”. However, we believe that there is no lost of security because of the presence of other security functional requirements, as explained in the following.

Due to the Commercial Grade OS Requirement Set “1.2.1.3” and “1.2.1.4” requirements, the integrity of transmitted data is ensured. Typically, as soon as the transmitted data is received, it is consumed straight away. Also, before the received data is consumed by writing, the corresponding “write” access permission would have been granted according to:

* the MIC “no write up” [i.e. Action a)] rule based on the integrity level (IL) of the writer subject and the IL of the target file directory and/or the target file object;
* the DAC rule that the writer subject is granted the “write” access right to the target file directory and/or the target file object.

In other words, the actual integrity level of the transmitted data is irrelevant.

Furthermore, as explained in the justification text for addressing the Commercial Grade OS Requirement Set “6.2.1.1” requirement, if the received data is written to a new target file object, then the Windows OS associates DAC and MIC security attributes to the new target file object. Specifically, the following are effective in the Windows OS.

In the case where

* the target named file is new;
* the subject does not provide a SYSTEM\_MANDATORY\_LABEL\_ACE ACE for the file;
* the parent NTFS directory object containing the target named file does not have an inheritable SYSTEM\_MANDATORY\_LABEL\_ACE ACE for its child file objects,

the target named file is assigned the default SYSTEM\_MANDATORY\_LABEL\_ACE ACE of integrity level value of MandatoryLevelMedium (2) and the write (SYSTEM\_MANDATORY\_LABEL\_NO\_ WRITE \_UP) access operation class.

In the case where

* the target named file is new;
* the subject does not provide a SYSTEM\_MANDATORY\_LABEL\_ACE ACE for the file;
* the parent NTFS directory object containing the target named file has an inheritable SYSTEM\_MANDATORY\_LABEL\_ACE ACE for its child file objects,

the target named file inherits the inheritable SYSTEM\_MANDATORY\_LABEL\_ACE ACE from its parent NTFS directory object.

In the case where

* the target named file is new;
* the subject does not provide a DACL for the file;
* the parent NTFS directory object containing the target named file does not have an inheritable DAC policy enforcement ACE for its child file objects,

the DACL for the target named file is the subject’s default DACL.

In the case where

* the target named file is new;
* the subject does not provide a DACL for the file;
* the parent NTFS directory object containing the target named file has an inheritable DAC policy enforcement ACE for its child file objects,

the target named file inherits the inheritable DAC policy enforcement ACE from its parent NTFS directory object.

As a result, the above has shown that there is no lost of security because of the presence of other security functional requirements.

## Addressing 2.2.1.8 “Function on the protection of integrity of user data transmitted within the OS, such as inter-process communication data shall be provided: The system shall ensure that data with lower integrity cannot be inserted into and cannot replace the data with higher integrity”

Due to the Commercial Grade OS Requirement Set “1.2.1.3” and “1.2.1.4” requirements, the Windows OS also addresses the first part of the “2.2.1.8” requirement, namely “Function on the protection of integrity of user data transmitted within the OS, such as inter-process communication data shall be provided”.

The second part of the “2.2.1.8” requirement, namely “The system shall ensure that data with lower integrity cannot be inserted into and cannot replace the data with higher integrity”, treats similar security concern(s) as the Commercial Grade OS Requirement Set “2.2.1.5” requirement “The OS shall permit an information flow among subjects and objects based on a specific set of rules”. The Commercial Grade OS Requirement Set requires the following specific set of rules for information flows among subjects and objects in a hierarchical integrity attributes scheme.

1. If the integrity label of the subject is greater than or equal to the integrity label of the object, then a write (the flow of information from the subject to the object) is permitted;
2. If the integrity label of the object is greater than or equal to the integrity label of the subject, then a read (the flow of information from the object to the subject) is permitted;
3. If the information flow is between objects, the integrity label of the source object must be greater than or equal to the integrity label of the destination object.

The Windows OS also addresses the second part of the “2.2.1.8” requirement because it is equivalent to Action c) “If the information flow is between objects, the integrity label of the source object must be greater than or equal to the integrity label of the destination object” of the hierarchical integrity attributes schemes of the Commercial Grade OS Requirement Set “2.2.1.5” requirement”.

Please see the justification text for addressing the Commercial Grade OS Requirement Set “2.2.1.5” requirement for detail.

## Addressing 2.2.1.9 “MIC based integrity protection of cross-network user data residing in file objects shall be realized by taking into account of the consistency of security attributes between subjects and file/directory objects of each OS computer in a network environment”

The Windows OS addresses the “2.2.1.9” requirement. The Mandatory Integrity Control (MIC) policy is enforced regardless of the local or remote access request mode. The access requester subject is associated with either a Windows OS local user account or a Windows OS domain wide user account. Please see the justification text for addressing the Commercial Grade OS Requirement Set “2.2.1.1” requirement for detail.

# Meeting the “Access Control Polices Mandatory Access Control Policy for User File/Directory Objects Functional Requirements”

There are 6 individual functional requirements under the heading of “Access Control Polices Mandatory Access Control Policy for User File/Directory Objects Functional Requirements”. They are listed as “2.3.1.n”, where n = 1, 2, 3, 4, 5, and 6.

## Addressing 2.3.1.1 “The OS shall adopt a labeling method to set sensitive labels for subjects and user file/directory objects”

The Windows OS addresses the “2.3.1.1” requirement for Windows OS file and directory objects because of the availability of the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” in the Windows OS. Please see the following “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” section for detail.

### Poor man’s Mandatory Access Control (MAC) for user file and directory objects

A Mandatory Access Control (MAC) policy is characterized by the following rules with hierarchical sensitivity levels which are associated with subjects and specific file/directory objects:

1. “No read up”
   * A subject of lower sensitivity level (SL) is not permitted to read from a file/directory object of higher sensitivity level (SL).
2. “No write down”
   * A subject of higher sensitivity level (SL) is not permitted to write to a file/directory object of lower sensitivity level (SL).
3. “Only administrators are allowed to change a sensitivity level (SL) to another sensitivity level (SL)”.

As described in the following section of this paper, the concept of “creator owner rights” ACE is available in the exercise of the Windows OS access determination algorithm.

* “Addressing 2.1.2.1 “The OS shall allow only authorized administrators, object owners, and users with the DAC change authorization the ability to change the access permission associated with a named object””.

When a specific “creator owner rights” ACE does not grant the [WRITE\_DAC](http://msdn.microsoft.com/en-us/library/aa379607.aspx) or the [WRITE\_OWNER](http://msdn.microsoft.com/en-us/library/aa379607.aspx) standard access control element, a consequence of the corresponding “creator owner rights” ACE concept effectively removes the discretionary aspect from the Discretionary Access Control (DAC) policy.

Within a Windows OS NTFS volume, it is possible to simulate the above MAC “no read up” and “no write up” rules by configuring a specific file directory hierarchy, rooted at say “User Data”, as follows, due to the “creator owner rights” ACE concept.

* For each user SL n, define a user group called “SL n Group” so that the membership of “SL n Group” consists of exactly those user accounts of SL n:
  + For example,
    - the group “SECRET Users” consists of only user accounts of the SECRET sensitivity level as its members;
    - the group “UNCLASSIFIED Users” consists of only user accounts of the UNCLASSIFIED sensitivity level as its members.
* For each file/directory SL p, define a file directory called “SL p Data”, under the “User Data” directory, so that the “SL p Data” directory contains exactly those files or subdirectories of SL p:
  + For example,
    - the “SECRET Data” directory contains only files or subdirectories of the SECRET sensitivity level;
    - the “UNCLASSIFIED Data” directory contains only files or subdirectories of the UNCLASSIFIED sensitivity level.
* At the “User Data” directory, assign the following.
  + The Inheritable “creator owner rights” ACE which grants only the [DELETE](http://msdn.microsoft.com/en-us/library/aa379607.aspx) standard access control element to the SID “S-1-3-4” which contains the SECURITY\_CREATOR\_OWNER\_RIGHTS\_RID (4) under the SECURITY\_CREATOR\_SID\_AUTHORITY (S-1-3).
  + The inheritable ACCESS\_ALLOWED\_ACE which grants only the generic read and execute access rights to “SL n Group”, for each SL n.
    - For example,
      * The inheritable ACCESS\_ALLOWED\_ACE which grants only the generic read and execute access rights to the “SECRET Users” group;
      * The inheritable ACCESS\_ALLOWED\_ACE which grants only the generic read and execute access rights to the “UNCLASSIFIED Users” group.
* For each SL p, at the “SL p Data” directory, assign the following.
  + The inheritable ACCESS\_ALLOWED\_ACE which grants only the generic read and write access rights to “SL p Group”.
  + The inheritable ACCESS\_ALLOWED\_ACE which grants only the generic read access right to “SL q Group”, for each SL q, where q > p.

Assume that SECRET is the highest sensitivity level and UNCLASSIFIED is the lowest sensitivity level, we have the following examples.

* At the “SECRET Data” directory, assign
  + The inheritable ACCESS\_ALLOWED\_ACE which grants only the generic read and write access rights to the “SECRET Users” group.
* At the “UNCLASSIFIED Data” directory, assign
  + The inheritable ACCESS\_ALLOWED\_ACE which grants only the generic read and write access rights to the “UNCLASSIFIED Users” group.
  + The inheritable ACCESS\_ALLOWED\_ACE which grants only the generic read access right to the “SECRET Users” group.
  + The inheritable ACCESS\_ALLOWED\_ACE which grants only the generic read access right to “SL q Group”, for each SL q, where SECRET > q > UNCLASSIFIED.

Evidently, the MAC “no read up” and “no write up” rules are immediate consequence in the above security configuration of the “User Data” directory and its subdirectories.

The following diagram pictorially demonstrates the security configuration of the “User Data” directory and its subdirectories.



Finally, we note that the above security configuration for the “Poor man’s Mandatory Access Control (MAC)” is also applicable to the following objects:

* Windows OS Registry Key objects;
* Objects residing under Windows OS object manager kernel object directory objects;
* Active Directory objects and attributes.

## Addressing 2.3.1.2 “The sensitive labels constitute an attribute repository with a multi-level security model for user file/directory objects”

The Windows OS addresses the “2.3.1.2” requirement for Windows OS file and directory objects because of the availability of the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” in the Windows OS. Please see the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” section under Appendix B of this paper for detail.

## Addressing 2.3.1.3 “The OS shall enforce a Mandatory Access Control policy on untrusted subjects, named user file/directory objects and all operations that cause information to flow among them”

The Windows OS addresses the “2.3.1.3” requirement for Windows OS file and directory objects because of the availability of the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” in the Windows OS. Evidently, the security configuration for the “Poor man’s Mandatory Access Control (MAC)” is based on the following user groups and file system directories.

* The set of “SL n Group” of user groups, where n is a user sensitivity level (SL), and their memberships.
* The set of “SL p Data” file system directories, where p is a file/directory sensitivity level (SL).

Please see the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” section under Appendix B of this paper for detail.

## Addressing 2.3.1.4 “A uniform mandatory access control function for user file/directory objects shall be provided for the distributed OS running in a network environment”

The Windows OS addresses the “2.3.1.4” requirement for Windows OS file and directory objects because of the availability of the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” in the Windows OS. Evidently, the security configuration for the “Poor man’s Mandatory Access Control (MAC)” is based on the following user groups and file system directories.

* The set of “SL n Group” of user groups, where n is a user sensitivity level (SL), and their memberships.
* The set of “SL p Data” file system directories, where p is a file/directory sensitivity level (SL).

Members of a “SL n Group” of user groups could be either user accounts defined in the local Windows OS machine (i.e. Windows OS local user accounts) or user accounts defined in the Active Directory (i.e. Windows OS domain wide user accounts). In the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects”, the MAC policy is enforced regardless of the local or remote access request mode. The access requester subject is associated with either a Windows OS local user account or a Windows OS domain wide user account.

Please see the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” section under Appendix B of this paper for detail.

## Addressing 2.3.1.5 “MAC based confidentiality protection of cross-network user data residing in file objects shall be realized by taking into account of the consistency of security attributes between subjects and file/directory objects of each OS computer in a network environment”

The Windows OS addresses the “2.3.1.5” requirement for Windows OS file and directory objects because of the availability of the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” in the Windows OS. Evidently, the security configuration for the “Poor man’s Mandatory Access Control (MAC)” is based on the following user groups and file system directories.

* The set of “SL n Group” of user groups, where n is a user sensitivity level (SL), and their memberships.
* The set of “SL p Data” file system directories, where p is a file/directory sensitivity level (SL).

Members of a “SL n Group” of user groups could be either user accounts defined in the local Windows OS machine (i.e. Windows OS local user accounts) or user accounts defined in the Active Directory (i.e. Windows OS domain wide user accounts). In the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects”, the MAC policy is enforced regardless of the local or remote access request mode. The access requester subject is associated with either a Windows OS local user account or a Windows OS domain wide user account.

Please see the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” section under Appendix B of this paper for detail.

## Addressing 2.3.1.6 “The OS shall permit an information flow among subjects and user file/directory objects based on a specific set of rules based on hierarchical sensitivity levels”

The “2.3.1.6” requirement requires the following specific set of rules for information flows among subjects and user file/directory objects.

1. If the sensitivity label of the subject is greater than or equal to the sensitivity label of the object, then a read (the flow of information from the object to the subject) is permitted;
2. If the sensitivity label of the object is greater than or equal to the sensitivity label of the subject, then a write (the flow of information from the subject to the object) is permitted;
3. If the information flow is between objects, the sensitivity label of the destination object must be greater than or equal to the sensitivity label of the source object.

The Windows OS addresses the “2.3.1.6” requirement for Windows OS file and directory objects because of the availability of the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” in the Windows OS.

Please see the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” section under Appendix B of this paper for detail.

In the security configuration for the “Poor man’s Mandatory Access Control (MAC)” to control access to user file and directory objects, a “blind” write (i.e. writing without reading) is not possible. Therefore, the above b. rule of the “2.3.1.6” requirement is moot.

Additionally, the Windows OS does not implement a mechanism which allows a named object to cause an information flow to another named object without involving a subject who has been granted the necessary access rights in the acquisition of valid object handles to both named objects. Therefore, the above c. rule of the “2.3.1.6” requirement is moot.

# Meeting the “Access Control Polices Mandatory Access Control Policy for User File/Directory Objects Management Requirements”

There is 1 individual management requirement under the heading of “Access Control Polices Mandatory Access Control Policy for User File/Directory Objects Management Requirements”. It is listed as “2.3.2.1”.

## Addressing 2.3.2.1 “The sensitive labels for user file/directory objects and subjects of operating system should be established, maintained, and managed by authorized administrators”

The Windows OS addresses the “2.3.2.1” requirement for Windows OS file and directory objects because of the availability of the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” in the Windows OS. As the following are necessary established, maintained, and managed by an authorized administrator, the sensitive labels for file/directory objects and subjects are established, maintained, and managed by authorized administrators.

* The set of “SL n Group” of user groups, where n is a user sensitivity level (SL), and their memberships.
* The set of “SL p Data” file system directories, where p is a file/directory sensitivity level (SL).

Please see the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects” section under Appendix B of this paper for detail.

# Meeting the “Access Control Polices Mandatory Access Control Policy for User File/Directory Objects Audit Requirements”

There are 2 individual audit requirements under the heading of “Access Control Polices Mandatory Access Control Policy for User File/Directory Objects Audit Requirements”. They are listed as “2.3.3.n”, where n = 1, and 2.

## Addressing 2.3.3.1 “The OS shall provide the ability to audit all Mandatory Access Control policy decisions”

The Windows OS addresses the “2.3.3.1” requirement in the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects”.

The Windows OS generates the following [Event ID 4663](http://support.microsoft.com/kb/947226) audit record after it decides a subject’s requested access to a user file and directory object according to the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects”.

* [Event ID 4663](http://support.microsoft.com/kb/947226) “An attempt was made to access an object”.

## Addressing 2.3.3.2 “The OS shall provide the ability to audit the changing of subject and object sensitivity labels”

The Windows OS addresses the “2.3.3.2” requirement in the “Poor man’s Mandatory Access Control (MAC) for user file and directory objects”.

Recall that a user sensitivity label/level (SL) n is defined by the membership of the “SL n Group” user group for the user account of the SL n user. Therefore, the group membership change auditing, such as the following [Event ID 4728](http://support.microsoft.com/kb/947226) and [Event ID 4733](http://support.microsoft.com/kb/947226) audit records, is also interpreted as the subject sensitivity label/level change/configuration auditing.

* [Event ID 4728](http://support.microsoft.com/kb/947226) “A member was added to a security-enabled global group”;
* [Event ID 4733](http://support.microsoft.com/kb/947226) “A member was removed from a security-enabled local group”.

Recall that a file/directory object sensitivity label/level (SL) p is defined by the object’s residence in the “SL p Data” directory. Therefore, specific aspects of the file/directory object access audit, such as the following [Event ID 4663](http://support.microsoft.com/kb/947226) audit record, are also interpreted as the file/directory object sensitivity label/level change/configuration auditing.

* [Event ID 4663](http://support.microsoft.com/kb/947226) “An attempt was made to access an object”.

# Meeting Additional “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Functional Requirements”

The Commercial Grade OS Requirement Set already has 14 individual functional requirements under the heading of “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Functional Requirements”. They are listed as “3.1.1.n”, where n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, and 14.

There are 9 additional individual functional requirements under the heading of “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Functional Requirements” in this appendix. They are listed as “3.1.1.n”, where n = 15, 16, 17, 18, 19, 20, 21, 22, and 23.

## Addressing 3.1.1.15 “User authentication shall adopt a selection of password authentication, token based dynamic password authentication, biometric authentication, digital certificate authentication and others”

The “3.1.1.15” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.1.1” requirement “The OS shall require each user to be uniquely identified and successfully authenticated by means of a password before allowing any actions on behalf of that user”.

The Windows OS also addresses the “3.1.1.15” requirement because the Windows OS adopts password logon, smart card logon, and [RFC 2246 Transport Layer Security (TLS)](http://msdn2.microsoft.com/en-us/library/aa380123(VS.85).aspx) remote logon which is digital certificate based. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.1” requirement for detail.

## Addressing 3.1.1.16 “Before the establishment of an OS logon session, user identity shall be authenticated and the logon mechanism is not allowed to bypass the authentication mechanism”

The “3.1.1.16” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.1.1” requirement “The OS shall require each user to be uniquely identified and successfully authenticated by means of a password before allowing any actions on behalf of that user”.

The Windows OS also addresses the “3.1.1.16” requirement with the same minor exceptions in the case where an unauthenticated user needs to look up “public” information within the network. As explained in the “User authentication” sections of this paper, the logon machine, as implemented in the Windows OS Authentication Service (aka lsass.exe) cannot be bypassed for both interactive and network logon scenarios, except in the case where an unauthenticated user needs to look up “public” information within the network.

Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.1” requirement for detail.

## Addressing 3.1.1.17 “The OS shall realize authentication failure treatment through pre-definition of unsuccessful trial values for authentication (including the time threshold value)”

The “3.1.1.17” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.1.3” requirement “The OS authentication mechanism shall provide a specific set of capabilities”. The Commercial Grade OS Requirement Set requires the following specific set of capabilities:

1. for all administrator accounts, a delay such that there can be no more than ten attempts per minute;
2. for all other accounts, a delay such that there can be no more than twenty attempts per minute.

The Windows OS also addresses the “3.1.1.17” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.3” requirement for detail.

## Addressing 3.1.1.18 “The OS shall realize provision of treatment measures when the time threshold value (of 3.1.1.17) is reached”

The “3.1.1.18” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.1.3” requirement “The OS authentication mechanism shall provide a specific set of capabilities”. The Commercial Grade OS Requirement Set requires the following specific set of capabilities:

1. for all administrator accounts, a delay such that there can be no more than ten attempts per minute;
2. for all other accounts, a delay such that there can be no more than twenty attempts per minute.

The Windows OS also addresses the “3.1.1.18” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.3” requirement for detail.

## Addressing 3.1.1.19 “The OS shall realize authentication failure treatment through pre-definition of unsuccessful trial values for authentication (including number of trial times)”

The “3.1.1.19” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.1.5” requirement “The OS shall detect when an authorized administrator specified positive integer of consecutive unsuccessful authentication attempts occur related to any authorized user authentication process”.

The Windows OS also addresses the “3.1.1.19” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.5” requirement for detail.

## Addressing 3.1.1.20 “The OS shall realize provision of treatment measures when the number of trial times (of 3.1.1.19) is reached”

The “3.1.1.20” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.1.6” requirement “The OS shall perform a specific set of actions when the defined number of consecutive unsuccessful authentication attempts specified in “3.1.1.5” has been detected”. The Commercial Grade OS Requirement Set requires the following specific set of actions being performed by the OS:

1. for all administrator accounts, disable the account for an authorized administrator configurable time period;
2. for all other accounts, disable the user account until it is re-enabled by an authorized administrator;
3. for all disable accounts, respond with an “account disabled” message without attempting any type of authentication.

The Windows OS also addresses the “3.1.1.20” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.6” requirement for detail.

## Addressing 3.1.1.21 “The OS shall realize unique user identifier in the whole life cycle of the OS, and the consistency between user names or alternative names and UIDs when User name and user identifier (UID) are adopted by the OS user identification”

The “3.1.1.21” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.1.7” requirement “The OS shall maintain a specific list of security attributes belonging to individual users”. The Commercial Grade OS Requirement Set requires the following specific list of security attributes belonging to individual users being maintained by the OS:

1. unique user identity;
2. group memberships;
3. authentication data;
4. any other security-relevant authorizations or attributes (e.g. roles).

As shown in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement, the Windows OS maintains a list of security attributes associated with a user account. The list includes:

* [ATT\_OBJECT\_SID](http://msdn.microsoft.com/en-us/library/ms679024(VS.85).aspx) (which is the security identifier (SID) of the user account);
* [ATT\_SAM\_ACCOUNT\_NAME](http://msdn.microsoft.com/en-us/library/ms679635(VS.85).aspx);
* [ATT\_USER\_PRINCIPAL\_NAME](http://msdn.microsoft.com/en-us/library/ms680857(VS.85).aspx).

Hence, the Windows OS adopts user name and user identifier.

Also shown in the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement, the structure of a SID, in the Windows OS, is [a variable-length byte array that uniquely identifies a security principal (i.e. a user or computer account, or a group)](http://msdn.microsoft.com/en-us/library/cc230371.aspx). Due to the RID monotonic-increasing characteristic and the SID array structure, unique SIDs (as user identifiers) are maintained/realized in the whole life cycle of the Windows OS, as required.

The consistency between [ATT\_SAM\_ACCOUNT\_NAME](http://msdn.microsoft.com/en-us/library/ms679635(VS.85).aspx) and [ATT\_OBJECT\_SID](http://msdn.microsoft.com/en-us/library/ms679024(VS.85).aspx) and the consistency between [ATT\_USER\_PRINCIPAL\_NAME](http://msdn.microsoft.com/en-us/library/ms680857(VS.85).aspx) and [ATT\_OBJECT\_SID](http://msdn.microsoft.com/en-us/library/ms679024(VS.85).aspx) are ensured because, during a user account creation, the [ATT\_SAM\_ACCOUNT\_NAME](http://msdn.microsoft.com/en-us/library/ms679635(VS.85).aspx) and the [ATT\_USER\_PRINCIPAL\_NAME](http://msdn.microsoft.com/en-us/library/ms680857(VS.85).aspx) (if not generated from the [ATT\_SAM\_ACCOUNT\_NAME](http://msdn.microsoft.com/en-us/library/ms679635(VS.85).aspx) in the default manner) are checked that they are not currently being associated with a previously created (i.e. existing) user account.

Finally, the Windows OS also addresses the rest of the “3.1.1.21” requirement as it does for the Commercial Grade OS Requirement Set “3.1.1.7” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.7” requirement for detail.

## Addressing 3.1.1.22 “User subject binding function shall be designed and realized for users, who register with the OS by connecting user process with the owner, to ensure that the behavior of the user process can be traced back to all owners of the process”

The “3.1.1.22” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.1.8” requirement “The OS shall associate a specific list of user security attributes with subjects acting on behalf of that user”. The Commercial Grade OS Requirement Set requires the following specific list of user security attributes for associating with subjects acting on behalf of that user:

1. the unique user identity;
2. any group identity or identities;
3. any other security-relevant authorizations or attributes (e.g. roles).

The Windows OS also addresses the “3.1.1.22” requirement because the assignment of the user security attributes to the primary access token. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.8” requirement for detail.

## Addressing 3.1.1.23 “User subject binding function shall be designed and realized for users, who register with the OS by dynamically connecting the system process with the current service requester, to ensure that the behavior of the system process can be traced back to current service requester”

The “3.1.1.23” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.1.8” requirement “The OS shall associate a specific list of user security attributes with subjects acting on behalf of that user”. The Commercial Grade OS Requirement Set requires the following specific list of user security attributes for associating with subjects acting on behalf of that user:

1. the unique user identity;
2. any group identity or identities;
3. any other security-relevant authorizations or attributes (e.g. roles).

The Windows OS also addresses the “3.1.1.23” requirement because the assignment of the user security attributes to the impersonation access token. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.1.8” requirement for detail.

# Meeting Additional “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Audit Requirements”

The Commercial Grade OS Requirement Set already has 5 individual audit requirements under the heading of “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Audit Requirements”. They are listed as “3.1.3.n”, where n = 1, 2, 3, 4, and 5.

There are 4 additional individual functional requirements under the heading of “Identification and Authentication User Identification/Authentication, Attributes, Roles, and Re-Authentication Audit Requirements” in this appendix. They are listed as “3.1.3.n”, where n = 6, 7, 8, and 9.

## Addressing 3.1.3.6 “The OS shall point out the least types of recordable audit events, including the change of system database management -- distribution and setup of users ID”

The “3.1.3.6” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.3.1” requirement “The OS shall provide the ability to audit the creation and management of user accounts”.

The Windows OS also addresses the “3.1.3.6” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.3.1” requirement for detail.

## Addressing 3.1.3.7 “The OS shall point out the least types of recordable audit events, including the change of system database management -- the change of user account attributes”

The “3.1.3.7” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.3.2” requirement “The OS shall provide the ability to audit the initialization and modification of user security attributes”.

The Windows OS also addresses the “3.1.3.7” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.3.2” requirement for detail.

## Addressing 3.1.3.8 “The OS shall point out the least types of recordable audit events, including the change of user identities as the order of super-users”

Assuming that the change of user identities as the order of super-users means the assignment of a user account to an administrator group or to an administrative privilege, the “3.1.3.8” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.3.2” requirement “The OS shall provide the ability to audit the initialization and modification of user security attributes”.

Assuming that the change of user identities as the order of super-users means the assignment of a user account to an administrator group or to an administrative privilege, the Windows OS also addresses the “3.1.3.8” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.3.2” requirement for detail.

## Addressing 3.1.3.9 “The OS shall point out the least types of recordable audit events, including the success and failure of login for session establishment, operational system interfaces”

The “3.1.3.9” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.1.3.5” requirement “The OS shall provide the ability to audit all user attempts to identify and authenticate to the system”.

The Windows OS also addresses the “3.1.3.9” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.1.3.5” requirement for detail.

# Meeting Additional “Identification and Authentication User Interface Security Functional Requirements”

The Commercial Grade OS Requirement Set already has 8 individual functional requirements under the heading of “Identification and Authentication User Interface Security Functional Requirements”. They are listed as “3.2.1.n”, where n = 1, 2, 3, 4, 5, 6, 7, and 8.

There are 6 additional individual functional requirements under the heading of “Identification and Authentication User Interface Security Functional Requirements” in this appendix. They are listed as “3.2.1.n”, where n = 9, 10, 11, 12, 13, and 14.

## Addressing 3.2.1.9 “The OS shall provide a mechanism for locking user keyboard, and request user authentication for the keyboard unlocking process”

The “3.2.1.9” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.2.1.2” requirement “The OS shall allow user-initiated locking of an interactive session by performing a specific set of actions”. The Commercial Grade OS Requirement Set requires the following specific set of actions being performed by the OS:

1. Clearing or overwriting display devices, making the current contents unreadable;
2. Disabling any activity from the user’s data access/display devices other than unlocking the interactive session.

The Windows OS also addresses the “3.2.1.9” requirement.

Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.2.1.2” requirement for detail.

## Addressing 3.2.1.10 “The OS shall be able to limit the maximum quantity of concurrent sessions of system, and use default values as the limited number of quantity of session, based on the basic user identification”

The “3.2.1.10” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.2.1.4” requirement “The OS shall enforce an authorized administrator specified maximum number of concurrent interactive sessions per user”.

The Windows OS also addresses the “3.2.1.10” requirement using the same workaround. The workaround is necessary due to a potential denial of service (DOS) attack induced by this requirement in an environment, where a high reliability level of the underlying (wired or wireless) network transport medium is not guaranteed.

Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.2.1.4” requirement for detail.

## Addressing 3.2.1.11 “After exceeding the set unused time limit, the OS shall disconnect the session or re-authenticate the user, and the system shall provide a default value on time limit”

The “3.2.1.11” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.2.1.5” requirement “The OS shall lock an interactive session after an authorized administrator specified time interval of user inactivity by performing a specific set of actions”.

The Windows OS also addresses the “3.2.1.11” requirement, except that the Windows OS does not disconnect the session, but locks the session instead.

Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.2.1.5” requirement for detail.

In the case of local session interaction maintenance, we recommend the allowance of session locking instead of session disconnection when the inactivity time limit is reached, for improving the user responsiveness aspect in the overall performance consideration. Session disconnection should apply when the inactivity time limit is reached only in the case of remote session interaction.

## Addressing 3.2.1.12 “The OS shall provide a mechanism that can regulate who can access to the system in accordance with time, access methods, location, network address or ports, etc”

For the time aspect, the “3.2.1.12” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.2.1.6” requirement “The OS shall provide the ability to deny interactive session establishment based on time and day”.

For the time aspect, the Windows OS also addresses the “3.2.1.12” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.2.1.6” requirement for detail.

Due to the availability of the following system access rights which are assignable to a user account or a group, the Windows OS also addresses the “3.2.1.12” requirement for the access methods aspect, where the access methods consist of the local method or the remote method:

* local method:
  + “[SeInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx)”;
  + “[SeDenyInteractiveLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx)”;
* remote method:
  + “[SeNetworkLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx)”;
  + “[SeDenyNetworkLogonRight](http://msdn.microsoft.com/en-us/library/cc207141.aspx)”.

For the location or network address aspect, the Windows OS does not meet the “3.2.1.12” requirement because

* the location or network address of a client for user authentication is unauthenticated;
* both location and network address are subject to spoofing

within the framework of the Windows OS supporting authentication protocols.

For the port aspect, the Windows OS does not meet the “3.2.1.12” requirement because the authentication protocols of the Windows OS are conducted over fixed well-known and documented ports, by default. For example, the following are fixed ports adopted by the Windows OS for its services.

* [Port 88](http://support.microsoft.com/kb/832017) for the Kerberos KRB\_AS\_REQ.
* [Port 464](http://support.microsoft.com/kb/832017) for the Kerberos change/set password protocol.
* [Port 135](http://support.microsoft.com/kb/832017) for RPC to carry the NTLM or Kerberos PDUs.
* [Port 445](http://support.microsoft.com/kb/832017) for SMB to carry the NTLM or Kerberos PDUs.
* [Port 80](http://support.microsoft.com/kb/832017) for HTTP to carry the RFC 2617 Digest Access Protocol PDUs.
* [Port 443](http://support.microsoft.com/kb/832017) for HTTPS to carry the RFC 2246 Transport Layer Security (TLS) Protocol PDUs.

## Addressing 3.2.1.13 “After the successful login, the OS shall record and show users specific information items”

The “3.2.1.13” requirement requires the following specific information item to be recorded and shown to the logon user.

* Date, time, source, and information of last successful login;
* Failure information on authentication since the last successful login;
* Number of days to password expiration;
* The number of successful or unsuccessful events can be displayed as integer and timestamp list.

For the date, time and source aspect, the “3.2.1.13” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “3.2.1.7” requirement “Upon successful interactive session establishment, the OS shall display to the authorized user a specific set of user access history information elements”. The Commercial Grade OS Requirement Set requires the following specific set of user access history information elements to be displayed to the authorized user:

1. Date, time, and location of that user’s last successful interactive session establishment;
2. The number of unsuccessful (interactive session establishment) authentication attempts for that user since last successful interactive session establishment.

With the same minor exception regarding the last successful logon source (i.e. location) information, the Windows OS also addresses the “3.2.1.13” requirement for the following two aspects:

* the last successful logon data and time;
* the unsuccessful logon attempt number since the user’s last successful logon.

Please see the justification text for addressing the Commercial Grade OS Requirement Set “3.2.1.7” requirement for detail.

The aspect of the number of successful logon attempts in the “3.2.1.13” requirement seems odd as the “3.2.1.13” requirement does not specify when the OS should start counting the number of successful logon attempts. If the counting starts from the last successful logon attempt, then the counting would be trivial, namely the number zero is always the number of successful logon attempts since the user’s last successful logon.

The Windows OS additionally addresses the “3.2.1.13” requirement for the “number of days to password expiration” aspect because of the “[PasswordExpiryWarning](http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/regentry/58537.mspx?mfr=true)” policy. This policy specifies how long before a password expires that the Windows OS prompts the user to change the password. The default is 14 days. The “[PasswordExpiryWarning](http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/regentry/58537.mspx?mfr=true)” policy may be configured by an authorized administrator through the “[PasswordExpiryWarning](http://www.microsoft.com/technet/prodtechnol/windows2000serv/reskit/regentry/58537.mspx?mfr=true)” registry value under the following registry key”

* HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon.

## Addressing 3.2.1.14 “The OS shall be able enforce different password restriction and accout lockout policies in a user’s logging on authentication attempts based on different roles of the user”

Our understanding of what this requirement is saying is the following. A user may have different user roles such as Role 1 and Role 2 for logging onto the OS. For Role 1, the administrator assigns a Role 1 specific value (x1) to the positive integer of consecutive unsuccessful authentication attempts for enforcing the Commercial Grade OS Requirement Set “3.1.1.6” requirement, where the OS performs a specific action (such as disabling the corresponding logging on user account associated with Role 1 until it is re-enabled by an authorized administrator) when the defined number of consecutive unsuccessful authentication attempts has been detected. For Role 2, the administrator assigns a Role 2 specific value (x2, where x1 ≠ x2) to the positive integer of consecutive unsuccessful authentication attempts for enforcing the “3.1.1.6” requirement. Subsequently, the user would not be able to log on using Role 1 after x1 consecutive unsuccessful authentication attempts have been detected; and the user would not be able to log on using Role 2 after x2 consecutive unsuccessful authentication attempts have been detected.

Note that an operating system is supposed to handle the identity based authentication (through a user account) rather than the role based authentication (through a specific role). This is clear from the Commercial Grade OS Requirement Set “3.1.1.1” requirement, which states that “The OS shall require each user to be uniquely identified and successfully authenticated by means of a password before allowing any actions on behalf of that user”. As a result, in order to let a user log onto the OS using a specific role, the user must submit a user account name to the OS, before providing any logging on credentials (such as a valid password). Let us interpret roles as groups in the OS. The OS enumerates all the roles (in the form of group memberships) associated with the user account. The OS displays the enumerated roles (group memberships) to the user for his/her selection during his/her logging on attempt to the specified user account. The role (group membership) selection display is necessary as the user typically does not know what roles (group memberships) that his/her user account belongs to.

Unfortunately, most enterprise organization customers deem the display of roles (group memberships) associated with a user account to unauthenticated users as a security leakage of their private information. The users to whom the role (group membership) selection is displayed are necessarily unauthenticated, because the user authentication to the specific user account has not occurred completely. Therefore, we find this “3.2.1.14” requirement difficult to support in the OS if the user is given only one user account to logon to the OS.

In the following, we attempt to address the “3.2.1.14” requirement in the “fallback” situation where a user may have one or more separate user accounts for logging onto the OS. The different user accounts of the user belong to different groups (i.e. roles). The different groups (i.e. roles) may have different conditions to achieve a successful logon for the user. The user is required to remember his/her individual user accounts of different groups (i.e. roles). For example, the user could have been created User Account 1 and User Account 2 for logging onto the OS by the administrator, where User Account 1 and User Account 2 belong to Group 1 and Group 2 respectively. When attempting to logon to User Account 1, the logging on attempt is subject to the consecutive unsuccessful authentication attempt detection. Specifically, the OS performs a specific action (such as disabling User Account 1 until it is re-enabled by an authorized administrator) when the Group 1 / Role 1 specific number, namely x1, of consecutive unsuccessful authentication attempts has been detected. Similarly, when attempting to logon to User Account 2, the logging on attempt is also subject to the consecutive unsuccessful authentication attempt detection. Specifically, the OS performs a specific action (such as disabling User Account 2 until it is re-enabled by an authorized administrator) when the Group 2 / Role 2 specific number, namely x1, of consecutive unsuccessful authentication attempts has been detected.

In the following, we explain how Windows Server 2008 uses its [fine-grained password and account lockout policies](http://technet.microsoft.com/en-us/library/cc770842.aspx) to enable the enforcement of the “3.1.1.6” requirement, where the OS performs a specific action (such as disabling the corresponding logging on user account associated with a given group/role until it is re-enabled by an authorized administrator) when the group/role specific number of consecutive unsuccessful authentication attempts has been detected. By enabling the enforcement of the “3.1.1.6” requirement, we would have shown that Windows Server 2008 addresses the “3.2.1.14” requirement in the “fallback” situation where a user may have one or more separate user accounts for logging onto the OS.

Windows Server 2008 includes the following two object classes in the Active Directory Domain Services (AD DS) schema to store fine-grained password restriction and account lockout policies:

* Password Settings Container;
* Password Settings.

The Password Settings Container (PSC) object of the PSC object class is created by default under the System container in a Windows AD domain. It stores the Password Settings objects (PSOs) of the Password Settings object class for that domain.

A PSO has attributes for all the settings that can be defined in the Windows Default Domain Policy (except Kerberos settings). These settings correspond to attributes for the following password restriction and account lockout policy settings:

* Enforce password history ([msDS-PasswordHistoryLength](http://msdn.microsoft.com/en-us/library/cc220330(PROT.13).aspx))
  + It is used in the PSO to override the “password history length” domain policy described in the “Addressing 3.1.2.6 “The OS shall allow only authorized administrators the ability to initialize and modify authentication mechanism attributes”” section of this paper;
* Maximum password age ([msDS-MaximumPasswordAge](http://msdn.microsoft.com/en-us/library/cc220303(PROT.13).aspx))
  + It is used in the PSO to override the “maximum password age” domain policy described in the “Addressing 3.1.2.6 “The OS shall allow only authorized administrators the ability to initialize and modify authentication mechanism attributes”” section of this paper;
* Minimum password age ([msDS-MinimumPasswordAge](http://msdn.microsoft.com/en-us/library/cc220307(PROT.13).aspx))
  + It is used in the PSO to override the “minimum password age” domain policy described in the “Addressing 3.1.2.6 “The OS shall allow only authorized administrators the ability to initialize and modify authentication mechanism attributes”” section of this paper;
* Minimum password length ([msDS-MinimumPasswordLength](http://msdn.microsoft.com/en-us/library/cc220308(PROT.13).aspx))
  + It is used in the PSO to override the “minimum password length” domain policy described in the “Addressing 3.1.2.6 “The OS shall allow only authorized administrators the ability to initialize and modify authentication mechanism attributes”” section of this paper;
* Passwords must meet complexity requirements ([msDS-PasswordComplexityEnabled](http://msdn.microsoft.com/en-us/library/cc220329(PROT.13).aspx))
  + It is used in the PSO to override the “password complexity” domain policy described in the “Addressing 3.1.2.6 “The OS shall allow only authorized administrators the ability to initialize and modify authentication mechanism attributes”” section of this paper;
* Store passwords using reversible encryption ([msDS-PasswordReversibleEncryptionEnabled](http://msdn.microsoft.com/en-us/library/cc220331(PROT.13).aspx))
  + It is used in the PSO to override the the “Store passwords using reversible encryption” domain password policy described in the “Addressing 3.1.1.7 “The OS shall maintain a specific list of security attributes belonging to individual users””section of this paper;
* Account lockout duration ([msDS-LockoutDuration](http://msdn.microsoft.com/en-us/library/cc220297(PROT.13).aspx))
  + It is used in the PSO to override the “lockout duration” domain policy described in the “Addressing 3.1.2.6 “The OS shall allow only authorized administrators the ability to initialize and modify authentication mechanism attributes”” section of this paper;
* Account lockout threshold ([msDS-LockoutThreshold](http://msdn.microsoft.com/en-us/library/cc220298(PROT.13).aspx))
  + It is used in the PSO to override the “lockout threshold” domain policy described in the “Addressing 3.1.2.6 “The OS shall allow only authorized administrators the ability to initialize and modify authentication mechanism attributes”” section of this paper;
  + It defines the positive integer of consecutive unsuccessful authentication attempts for enforcing the “3.1.1.6” requirement;
* Reset account lockout counter after ([msDS-LockoutObservationWindow](http://msdn.microsoft.com/en-us/library/cc220299(PROT.13).aspx))
  + It is used in the PSO to override the “reset lockout counter after” domain policy described in the “Addressing 3.1.2.6 “The OS shall allow only authorized administrators the ability to initialize and modify authentication mechanism attributes”” section of this paper.

If the password restriction and account lockout policy settings of a PSO are applied to members of a group, then an authorized administrator needs to link the group to the [ATT\_MS\_DS\_PSO\_APPLIES\_TO](http://msdn.microsoft.com/en-us/library/cc220346(PROT.13).aspx) attribute of the PSO. Following [the steps for configuring a PSO](http://technet.microsoft.com/en-us/library/cc770842.aspx) by an authorized administrator, members of the group are subject to the password restriction and account lockout policy settings of the PSO, instead of the the domain password restriction and account lockout policy settings.

Let PSO 1 be the PSO that Group 1 is linked to and let PSO 2 be the PSO that Group 2 is linked to. When attempting to logon to User Account 1, which belongs to Group 1, the logging on attempt to User Account 1 is subject to the password restriction and account lockout policy settings of the PSO 1. When attempting to logon to User Account 2, which belongs to Group 2, the logging on attempt to User Account 2 is subject to the password restriction and account lockout policy settings of the PSO 2.

In addition, the Windows OS can be configured to require the logging on attempts for specific groups of user accounts to use smart cards (and hence enforcing a two factor authentication) while the logging on attempts for other groups of user accounts to use only passwords.

Consequently, the Windows OS addresses the “3.2.1.14” requirement in the “fallback” situation where a user may have to have one or more separate user accounts for logging onto the OS

# Meeting Additional “Information Protection Residual Information Functional Requirements”

The Commercial Grade OS Requirement Set already has 2 individual functional requirements under the heading of “Information Protection Residual Information Functional Requirements”. They are listed as “5.1.1.n”, where n = 1 and 2.

There is 1 additional individual functional requirement under the heading of “Information Protection Residual Information Functional Requirements” in this appendix. It is listed as “5.1.1.3”.

## Addressing 5.1.1.3 “The OS shall ensure the resources under dynamic allocation and management can be re-used, when the information security is maintained, through specific actions”

The “5.1.1.3” requirement requires the following specific actions.

* Ensure that unauthorized users can not search information content on recording medium that has been returned to the OS after use;
* Ensure that unauthorized users can not search other old information on the recording medium allocated by the OS”.

The “5.1.1.3” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “5.1.1.1” requirement “The OS shall ensure that any previous information content of a resource is made unavailable upon either the allocation of the resource to or the de-allocation of the resource from all non cryptographic objects”.

Assuming the searching attempt is mediated by the Windows OS rather than being conducted through a direct physical access to the disk, the Windows OS also addresses the “5.1.1.3” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “5.1.1.1” requirement for detail.

# Meeting Additional “Information Protection Resource Control Functional Requirements”

The Commercial Grade OS Requirement Set already has 1 individual functional requirement under the heading of “Information Protection Resource Control Functional Requirements” and it is the “5.2.1.1” requirement.

There are 7 additional individual functional requirements under the heading of “Information Protection Resource Control Functional Requirements” in this appendix. They are listed as “5.2.1.n”, where n = 2, 3, 4, 5, 6, 7, and 8.

## Addressing 5.2.1.2 “The OS shall ensure its normal running by some measures when the OS is under identified failure (i.e. degraded) conditions, where the OS shall check and report when the OS service level is lower than the predetermined minimum”

The Windows OS addresses the “5.2.1.2” requirement because of the availability of the [Windows OS Reliability and Performance Monitor](http://technet.microsoft.com/en-us/library/cc749325.aspx). The [Windows OS Reliability and Performance Monitor](http://technet.microsoft.com/en-us/library/cc749325.aspx) is described in the following “Windows OS Reliability and Performance Monitor” section.

### Windows OS Reliability and Performance Monitor

The [Windows OS Reliability and Performance Monitor](http://technet.microsoft.com/en-us/library/cc749325.aspx) allows authorized administrators to configure the specific system and service performance attributes so that their corresponding performance counters can be monitored. Additionally, the authorized administrators can also set threshold values for specific counters so that alerts can be raised (as a means for reporting to the authorized administrators) when the threshold values are reached or fallen below. The following lists specific performance objects where there are associated performance counters in the Windows OS.

* [ACS Policy Performance Object](http://technet.microsoft.com/en-us/library/cc785126.aspx)
  + The Quality of Service (QoS) ACS Policy performance object provides data about the operation of Microsoft Identity Local Policy Module. Network administrators monitor the counters provided by this object when the QoS Admission Control Service (ACS) or Resource Reservation Protocol (RSVP) is enabled on a subnet.
* [Active Server Pages Performance Object](http://technet.microsoft.com/en-us/library/cc738099.aspx)
  + The Active Server Pages performance object consists of counters that monitor errors, requests, sessions, and other data generated by Active Server Pages (ASP pages) applications.
* [Battery Status Performance Object](http://technet.microsoft.com/en-us/library/cc780145.aspx)
  + The Battery Status performance object reports the battery status on laptops and on other computers that use batteries, based on Windows Management Instrumentation (WMI).
* [Browser Performance Object](http://technet.microsoft.com/en-us/library/cc783280.aspx)
  + The Browser performance object consists of counters that measure the rates of announcements, enumerations, and other browser transmissions.
* [Cache Performance Object](http://technet.microsoft.com/en-us/library/cc783778.aspx)
  + The Cache performance object consists of counters that monitor the file system cache, which is an area of physical memory that stores recently used data as long as possible to permit access to the data without having to read from the disk.
* [Database Performance Object](http://technet.microsoft.com/en-us/library/cc728414.aspx)
  + The Database performance object consists of counters that describe the activity of the Extensible Storage Engine (ESE), which is a transacted database management system that stores all Microsoft Active Directory objects.
* [DHCP Server Performance Object](http://technet.microsoft.com/en-us/library/cc740015.aspx)
  + The DHCP Server performance object consists of counters that monitor DHCP service activities.
* [Distributed Transaction Coordinator Performance Object](http://technet.microsoft.com/en-us/library/cc738252.aspx)
  + The performance object provides statistics about Microsoft Distributed Transaction Coordinator (MS DTC), which is part of Component Services (formerly known as Microsoft Transaction Server) that coordinates external (two-phase) transactions. Among other things, MS DTC is used to coordinate two-phase transactions by Message Queuing.
* [DNS Performance Object](http://technet.microsoft.com/en-us/library/cc783397.aspx)
  + The Domain Name System (DNS) is an Internet and TCP/IP standard name service. The DNS performance object consists of counters for monitoring various areas of performance of the DNS service, which enables client computers on the Windows OS network to register and resolve DNS domain names.
* [Fax Service Performance Object](http://technet.microsoft.com/en-us/library/cc776420.aspx)
  + The Shared Fax Service performance object tracks fax activities while the Fax service is running.
* [FileReplicaConn Performance Object](http://technet.microsoft.com/en-us/library/cc784859.aspx)
  + The FileReplicaConn performance object displays performance statistics of the Replicaconn object that defines replica connections to Distributed File System (DFS) roots.
* [FileReplicaSet Performance Object](http://technet.microsoft.com/en-us/library/cc737739.aspx)
  + The FileReplicaSet performance object displays performance statistics of the Replicaset object that defines a replica set (one or more replicas, shared volumes or directories that store duplicates of the contents of an original share). The object reports statistics for the computer that constitutes the original share.
* DFS Replication Service Volume Performance Object
  + The DFS Replication Service Volume performance object provides DFS Replication Service volume performance counters such as the number of update sequence number (USN) journal records that were read by the DFS Replication service, the number of update sequence number (USN) journal records that were processed by the DFS Replication service, and more.
* DFS Replicated Folders Performance Object
  + The DFS Replicated Folders performance object provides DFS Replicated Folders performance counters such as the number of files and folders in this replicated folder that were moved to the Conflict and Deleted folder by the DFS Replication service, the total size (in bytes) of the files and folders in this replicated folder that were moved to the Conflict and Deleted folder by the DFS Replication service, and more.
* DFS Replication Connections Performance Object
  + The DFS Replication Connections performance object provides DFS Replication Connections performance counters such as the total number of bytes received on a connection, the number of files that were received on a connection, and more.
* [FTP Service Performance Object](http://technet.microsoft.com/en-us/library/cc778510.aspx)
  + The FTP Service performance object consists of counters that show data about the Internet Information Services (IIS) FTP service.
* [Http Indexing Service Performance Object](http://technet.microsoft.com/en-us/library/cc757155.aspx);
  + The Http Indexing Service object reports statistics about queries run by the Indexing Service through Internet Information Services (IIS). These counters only report values when Indexing Service queries are executed through the Web server.
* [IAS Accounting Clients Performance Object](http://technet.microsoft.com/en-us/library/cc739238.aspx)
  + The IAS Accounting Clients performance object is installed by the Internet Authentication Service (IAS). IAS uses the Remote Authentication Dial-In User Service (RADIUS) protocol to perform remote authentication. IAS performance objects report activities for servers or clients, including user authentication, authorization, and accounting (usage).
* [IAS Accounting Proxy Performance Object](http://technet.microsoft.com/en-us/library/cc783879.aspx)
  + The IAS Accounting Proxy performance object is installed by Internet Authentication Service (IAS). IAS performance objects report activities for servers or clients, including user authentication, authorization, and accounting (usage).
* [IAS Accounting Server Performance Object](http://technet.microsoft.com/en-us/library/cc758238.aspx)
  + The IAS Accounting Server performance object is installed by Internet Authentication Service (IAS). IAS performance objects report activities for servers or clients, including user authentication, authorization, and accounting (usage).
* [IAS Authentication Clients Performance Object](http://technet.microsoft.com/en-us/library/cc786329.aspx)
  + The IAS Authentication Clients performance object is installed by Internet Authentication Service (IAS). IAS performance objects report activities for servers or clients, including user authentication, authorization, and accounting (usage).
* [IAS Authentication Proxy Performance Object](http://technet.microsoft.com/en-us/library/cc756933.aspx)
  + The IAS Authentication Proxy performance object is installed by Internet Authentication Service (IAS). IAS performance objects report activities for servers or clients, including user authentication, authorization, and accounting (use).
* [IAS Authentication Server Performance Object](http://technet.microsoft.com/en-us/library/cc780403.aspx)
  + The IAS Authentication Server performance object is installed by Internet Authentication Service (IAS). IAS performance objects report activities for servers or clients, including user authentication, authorization, and accounting (usage).
* [IAS Remote Accounting Servers Performance Object](http://technet.microsoft.com/en-us/library/cc728456.aspx)
  + The IAS Remote Accounting Servers performance object is installed by Internet Authentication Service (IAS). IAS performance objects report activities for servers or clients, including user authentication, authorization, and accounting (use).
* [IAS Remote Authentication Servers Performance Object](http://technet.microsoft.com/en-us/library/cc779355.aspx)
  + The IAS Remote Authentication Servers performance object is installed by Internet Authentication Service (IAS). IAS performance objects report activities for servers or clients, including user authentication, authorization, and accounting (usage).
* [ICMP Performance Object](http://technet.microsoft.com/en-us/library/cc784301.aspx)
  + The ICMP performance object consists of counters that measure the rates at which Internet Control Message Protocol (ICMP) messages are sent and received by using ICMP. It also includes counters that monitor ICMP errors.
* [Image Performance Object](http://technet.microsoft.com/en-us/library/cc787099.aspx);
  + The Image performance object consists of counters that monitor virtual address usage of images executed by processes on the computer. This object contains “costly” counters for which data collection is typically either processor-intensive or memory-intensive.
* [Indexing Service Performance Object](http://technet.microsoft.com/en-us/library/cc740022.aspx)
  + The Indexing Service performance object reports statistics about the creation of indexes, and about the merging and querying of indexes by the Windows OS Indexing Service.
* [Indexing Service Filter Performance Object](http://technet.microsoft.com/en-us/library/cc787539.aspx)
  + The Indexing Service Filter object monitors the filtering activities of the Windows OS Indexing Service.
* [Internet Information Services Global Performance Object](http://technet.microsoft.com/en-us/library/cc784792.aspx)
  + The Internet Information Services Global performance object consists of counters that apply jointly to the Hypertext Transport Protocol (HTTP) service and File Transfer Protocol (FTP) service.
* Internet Information Services [Windows Process Activation Service (WAS)](http://technet.microsoft.com/en-us/library/cc735229.aspx) Worker Processes Performance Object
  + The Internet Information Services WAS Worker Processes performance object provides the Internet Information Services WAS Worker Processes performance counters such as the number of currently active listener channels in a worker process, the number of currently active protocol handlers in a worker process, and more.
* Internet Information Services Web Admin Service HTTP Request Processing Performance Object
  + The Internet Information Services Web Admin Service HTTP Request Processing performance object provides the Internet Information Services Web HTTP Request Processing performance counters such as the number of HTTP requests served by a worker process, the number of threads available to process HTTP requests in a worker process, and more.
* [IP Performance Object](http://technet.microsoft.com/en-us/library/cc736621.aspx)
  + The IP performance object includes counters that describe the rates at which Internet Protocol (IP) datagrams are transmitted by a computer using the IP protocol. It also describes various IP error counts.
* [Job Object Performance Object](http://technet.microsoft.com/en-us/library/cc759530.aspx)
  + The Job Object performance object reports the accounting and processor usage data collected by each active job object. A job object provides a means of managing groups of processes with respect to their processor usage and other factors.
* [Job Object Details Performance Object](http://technet.microsoft.com/en-us/library/cc737859.aspx)
  + The Job Object Details performance object counters show detailed performance information about the active processes that make up a job objects.
* [LogicalDisk Performance Object](http://technet.microsoft.com/en-us/library/cc784596.aspx)
  + The LogicalDisk performance object consists of counters that monitor logical partitions of hard or fixed disk drives.
* [Memory Performance Object](http://technet.microsoft.com/en-us/library/cc778082.aspx)
  + The Memory performance object consists of counters that describe the behavior of physical and virtual memory on the computer. Physical memory is the amount of RAM on the computer. Virtual memory consists of space in physical memory and on disk. Many of the memory counters monitor paging, which is the movement of pages of code and data between disk and physical memory. Excessive paging is a symptom of a memory shortage and can cause delays that interfere with all system processes.
* [MSMQ Queue Performance Object](http://technet.microsoft.com/en-us/library/cc728409.aspx)
  + The MSMQ Queue performance object monitors message statistics for a specified queue. There are instances for each queue on the computer. In addition, if a Message Queuing–based application has a private queue open on another computer, an instance for that queue is also available. The computer queue instance represents the computer's source journal and dead letter queues.
* [MSMQ Service Performance Object](http://technet.microsoft.com/en-us/library/cc786045.aspx)
  + The MSMQ Service performance object monitors session and message statistics for a specified computer that is running Message Queuing.
* [MSMQ Session Performance Object](http://technet.microsoft.com/en-us/library/cc782511.aspx)
  + The MSMQ Session performance object monitors statistics about active sessions between computers running Message Queuing.
* [NBT Connection Performance Object](http://technet.microsoft.com/en-us/library/cc756753.aspx)
  + The NBT Connection performance object consists of counters that measure the rates at which bytes are sent and received over the NetBIOS over a TCP/IP (NetBT or NBT) connection between the local computer and a remote computer. The NetBT or NBT protocol provides NetBIOS support for the TCP/IP protocol. The NBT connection is identified by the name of the remote computer.
* [Network Interface Performance Object](http://technet.microsoft.com/en-us/library/cc757160.aspx)
  + The Network Interface performance object consists of counters that measure the rates at which bytes and packets are sent and received over a TCP/IP connection. It includes counters that monitor connection errors. The Network Interface counters display data about the network adapters on the server computer. The first instance of the Network Interface object (Instance 1) that is seen in System Monitor represents the loopback. The loopback is a local path through the protocol driver and the network adapter. All other instances represent installed network adapters.
* [NNTP Commands Performance Object](http://technet.microsoft.com/en-us/library/cc758614.aspx)
  + The NNTP Commands performance object includes counters for all Network News Transport Protocol (NNTP) commands processed by the NNTP service, which is installed as an optional component of Internet Information Services. NNTP counters are started for each virtual server instance and are reset to zero when the virtual server is started. A single NNTP service can have many virtual servers (also called sites).
* [NNTP Server Performance Object](http://technet.microsoft.com/en-us/library/cc776226.aspx)
  + The NNTP Server performance object monitors posting, authentication, and connection activity on a Network News Transport Protocol (NNTP) Server.
* [NTDS Performance Object](http://technet.microsoft.com/en-us/library/cc779676.aspx)
  + The NTDS object consists of counters that provide statistics about the activity of the Active Directory directory service. The object provides information about the following:
    - The Name Service Provider Interface (NSPI), used to facilitate communication between Active Directory and Exchange Directory Service (XDS).
    - The Local Security Authority (LSA), a protected subsystem that maintains security for the local computer.
    - The Security Accounts Manager (SAM) interface, which provides compatibility between Windows Server 2008, Windows Server 2003, Windows 2000, and Windows NT 4.0 domains.
    - The Lightweight Directory Access Protocol (LDAP) interface, which provides the application programming interface (API) for LDAP clients and exposes the Active Directory Services Interface (ADSI) so additional applications may be written that can talk to Active Directory.
    - The Directory System Agent (DSA), the Active Directory process that runs on each domain controller and manages all the directory service functions.
    - The Knowledge Consistency Checker (KCC), an Active Directory component that is responsible for generating the replication topology between domain controllers.
* [Objects Performance Object](http://technet.microsoft.com/en-us/library/cc785200.aspx)
  + The Objects performance object consists of counters that monitor logical objects in the system, such as processes, threads, mutexes, and semaphores. This information can be used to detect the unnecessary consumption of computer resources. Each object requires memory to store basic information about the object.
* [Paging File Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)
  + The Paging File performance object consists of counters that monitor the paging file(s) on the computer. The paging file is a reserved space on disk that backs up committed physical memory on the computer.
* [PBServer Monitor Performance Object](http://technet.microsoft.com/en-us/library/cc781378.aspx)
  + The PBServer Monitor performance object monitors activity on the phone book server. Phone Book Service is a comprehensive phone book management system, and is a component of Connection Point Services. Phone Book Service provides a central location for uploading and distributing access numbers for both public and private networks. A few of the counters, such as Error Hits, can be used to detect errors.
* [Physical Disk Performance Object](http://technet.microsoft.com/en-us/library/cc776376.aspx)
  + The PhysicalDisk performance object consists of counters that monitor hard or fixed disk drives. Disks are used to store file, program, and paging data. Disks are also read for retrieving the stored or paged items, and are written to record changes to them. The values of physical disk counters are sums of the values of the logical disks (or partitions) into which they are divided.
* [Print Queue Performance Object](http://technet.microsoft.com/en-us/library/cc782747.aspx)
  + The Print Queue performance object is provided by Windows Server family printing features to track activity in the server print queue. The Print Queue counters are reset when either the print server or the spooler service is restarted.
* [Process Performance Object](http://technet.microsoft.com/en-us/library/cc780836.aspx)
  + The Process performance object consists of counters that monitor running application program and system processes. All the threads in a process share the same address space and have access to the same data.
* [Process Address Space Performance Object](http://technet.microsoft.com/en-us/library/cc738423.aspx)
  + The Process Address Space performance object consists of counters that monitor memory allocation and use for a selected process. This object contains “costly” counters for which data collection is typically either processor-intensive or memory-intensive.
* [Processor Performance Object](http://technet.microsoft.com/en-us/library/cc786359.aspx)
  + The Processor performance object consists of counters that measure aspects of processor activity. The processor is the part of the computer that performs arithmetic and logical computations, initiates operations on peripherals, and runs the threads of processes. A computer can have multiple processors. The Processor object represents each processor as an instance of the object.
* [PSched Flow Performance Object](http://technet.microsoft.com/en-us/library/cc739640.aspx)
  + The PSched Flow Object shows statistics from the [packet scheduler](http://msdn.microsoft.com/en-us/library/aa373989(VS.85).aspx).
* [PSched Pipe Performance Object](http://technet.microsoft.com/en-us/library/cc778734.aspx)
  + The Packet Scheduler (PSched) Pipe performance object reports statistics from the packet scheduler pipe. On a local area network (LAN) adapter, the pipe represents the network interface card, and on a wide area network (WAN) adapter, the pipe represents each WAN line.
* [RAS Port Performance Object](http://technet.microsoft.com/en-us/library/cc738481.aspx)
  + The RAS Port performance object consists of counters that monitor individual remote access service (RAS) ports of the RAS device on the system.
* [RAS Total Performance Object](http://technet.microsoft.com/en-us/library/cc738450.aspx)
  + The RAS Total performance object consists of counters that combine values for all ports of the remote access service (RAS) device on the computer.
* [Redirector Performance Object](http://technet.microsoft.com/en-us/library/cc783876.aspx)
  + The Redirector performance object consists of counters that monitor server message block (SMB) network connections originating at the local computer.
* [Server Performance Object](http://technet.microsoft.com/en-us/library/cc757730.aspx)
  + The Server performance object consists of counters that measure server message block (SMB) communication between the local computer and the network.
* [Server Work Queues Performance Object](http://technet.microsoft.com/en-us/library/cc787885.aspx)
  + The Server Work Queues performance object consists of counters that monitor the length of queues (for handling the SMB processing) and objects in the queues (for handling the SMB processing).
* [SMTP Server Performance Object](http://technet.microsoft.com/en-us/library/cc736659.aspx)
  + The SMTP Server performance object consists of counters that describe the activity of the Exchange NTFS store driver, which is responsible for storing queued messages that are being processed by the Simple Mail Transfer Protocol (SMTP) service.
* [SMTP NTFS Store Driver Performance Object](http://technet.microsoft.com/en-us/library/cc739332.aspx)
  + The SMTP NTFS Store Driver performance object consists of counters that describe the activity of the Exchange NTFS store driver, which is responsible for message transport to and from the message store. The Simple Mail Transport Protocol (SMTP) is the protocol in the TCP/IP suite that directs the exchange of electronic mail.
* [System Performance Object](http://technet.microsoft.com/en-us/library/cc737309.aspx)
  + The System performance object consists of counters that apply to more than one component of the computer. The data collected by the system counters is derived from activity in the processor, memory, or disk subsystems.
* [Telephony Performance Object](http://technet.microsoft.com/en-us/library/cc783899.aspx)
  + The Telephony performance object provides counters for monitoring telephone-communication activity on a computer that is running a Windows Server family operating system.
* [TCP Performance Object](http://technet.microsoft.com/en-us/library/cc778758.aspx)
  + The TCP performance object consists of counters that measure the rates at which Transmission Control Protocol (TCP) segments are sent and received using TCP. It includes counters that monitor the number of TCP connections in each TCP connection state.
* Windows OS Terminal Services Summary Performance Object
  + The Windows OS Terminal Services Summary performance object provides the Terminal Services performance summary counters such as the number of active Terminal Services sessions, the number of inactive Terminal Services sessions, and more.
* [Terminal Services Performance Object](http://technet.microsoft.com/en-us/library/cc738439.aspx)
  + The Terminal Services object provides per-session resource statistics such as packet and frame transmission data, errors, and cache activity for Terminal Services.
* [Terminal Services Session Performance Object](http://technet.microsoft.com/en-us/library/cc759438.aspx)
  + The Terminal Services Session performance object provides per-session resource statistics, such as packet and frame transmission data, errors, and cache activity for Terminal Services.
* Windows OS [Terminal Services Gateway (TS Gateway)](http://technet.microsoft.com/en-us/library/cc731264.aspx) Performance Object
  + The Windows OS TS Gateway (TSG) performance object provides the TS Gateway performance counters such as the number of active/inactive connections to the TSG server at any given moment, the number of requests that failed due to insufficient connection authorization privilege, and more.
* [Thread Performance Object](http://technet.microsoft.com/en-us/library/cc737197.aspx)
  + The Thread performance object consists of counters that measure aspects of thread behavior. A thread is the basic object that executes instructions on a processor. All running processes have at least one thread.
* [Thread Details Performance Object](http://technet.microsoft.com/en-us/library/cc779381.aspx)
  + The Thread Details performance object consists of counters that measure aspects of thread behavior that are difficult or time-consuming to collect. These counters are distinguished from those in the Thread performance object by their high overhead. This object contains costly counters, for which data collection is typically either processor-intensive or memory-intensive.
* [UDP Performance Object](http://technet.microsoft.com/en-us/library/cc728379.aspx)
  + The UDP performance object consists of counters that measure the rates at which User Data Protocol (UDP) datagrams are sent and received using the UDP. It includes counters that monitor UDP errors.
* [Web Service Performance Object](http://technet.microsoft.com/en-us/library/cc783067.aspx)
  + The Web Service performance object counters (installed with Internet Information Services) monitor file transfer rates, bandwidth usage, connection rates, errors, and numbers and types of users.
* [Web Service Cache Performance Object](http://technet.microsoft.com/en-us/library/cc775855.aspx)
  + The Web Service Cache object counters installed with Internet Information Services monitor file transfer rates, bandwidth usage, and connection rates.
* [Windows Media Station Service Performance Object](http://technet.microsoft.com/en-us/library/cc759537.aspx)
  + The Windows Media Station Service performance object consists of counters that provide statistics about the Windows Media Station service, which provides multicasting, distribution, and storage functions for Advanced Systems Format (ASF) streams.
* [Windows Media Unicast Service Performance Object](http://technet.microsoft.com/en-us/library/cc728118.aspx)
  + The Windows Media Unicast Service performance object consists of counters that provide statistics about the Windows Media Unicast service, which provides unicasting functions for Advanced Systems Format (ASF) streams.
* [WINS Server Performance Object](http://technet.microsoft.com/en-us/library/cc738943.aspx)
  + The WINS Server performance object reports the activity of a computer that is running Windows Internet Name Service (WINS). This object's counters are cleared each time you start and stop the service. WINS provides a dynamic database service that can register and resolve NetBIOS names to the IP addresses that are used on the Windows OS network. The Windows Server family provides WINS, which enables the server computer to act as a NetBIOS name server and to register and resolve names for WINS-enabled client computers on the Windows OS network, as described in the NetBIOS over TCP/IP standards.
* [WMI Objects Performance Object](http://technet.microsoft.com/en-us/library/cc783954.aspx)
  + The WMI Objects performance object consists of counters that describe the activity of Windows Management Instrumentation (WMI) objects. WMI has a reverse adapter that makes WMI performance data visible in System Monitor. WMI performance data only appears when the reverse adapter for WMI is installed.
* [WSRM Policy Performance Object](http://technet.microsoft.com/en-us/library/cc787138.aspx)
  + The WSRM Policy performance object, installed with Windows System Resource Manager (WSRM), consists of a counter that provides the percentage of CPU cycles of the entire system that are used by the current resource-allocation policy, since the time the policy was set to manage the behavior of WSRM. (The resource-allocation policy is an ordered list of resource allocations that specifies the management behavior of WSRM at a given time.) Each WSRM Policy object will have one instance whose name is the same as the current active policy.
* [WSRM Process Performance Object](http://technet.microsoft.com/en-us/library/cc776704.aspx)
  + The WSRM Process performance object, installed with Windows System Resource Manager (WSRM), consists of a counter that provides the percentage of total managed CPU cycles used by a particular process, since the time that the resource-allocation policy was set to manage the behavior of WSRM. (The resource-allocation policy is an ordered list of resource allocations that specifies the management behavior of WSRM at a given time.) One instance of this object exists for every managed process. Note that the “Total Managed CPU” cycles is the sum of the CPU cycles for all the managed processes.
* [WSRM Process Matching Criteria Performance Object](http://technet.microsoft.com/en-us/library/cc758530.aspx)
  + The WSRM process-matching criteria performance object, installed with Windows System Resource Manager (WSRM), consists of counters that monitor the CPU usage and memory usage of the processes matched by the process-matching criteria. The criteria are included in the managing resource-allocation policy. Each object will have as many instances as the number of process-matching criteria within the current active policy.
* [ACS/RSVP Interfaces Performance Object](http://technet.microsoft.com/en-us/library/cc757163.aspx)
  + ACS/RSVP Interfaces performance object counters report statistics about Resource Reservation Protocol (RSVP) activity, flow, bandwidth, RSVP messages sent, and so on for a specific interface or IP address. The ACS/RSVP Interfaces object is available if QoS Admission Control Service (QoS ACS) or RSVP is running and there is one object for each IP address on the system.
* [ACS/RSVP Service Performance Object](http://technet.microsoft.com/en-us/library/cc759645.aspx)
  + The ACS/RSVP Service performance object reports statistics about the Resource Reservation Protocol (RSVP) service and about local applications that use Quality of Service (QoS). The ACS/RSVP Service object is available even if the RSVP service is not running.
* [Network Segment Performance Object](http://technet.microsoft.com/en-us/library/cc737622.aspx)
  + The Network Segment performance object provides network statistics for the local network segment.
* Virtual Machine BIOS Performance Object
  + The Virtual Machine BIOS performance object provides BIOS performance counters for a virtual machine.
* Virtual Machine Ethernet Controller Performance Object
  + The Virtual Machine Ethernet Controller performance object provides Ethernet controller performance counters for a virtual machine.
* Virtual Machine IDE Controller Performance Object
  + The Virtual Machine IDE Controller performance object provides IDE controller performance counters for a virtual machine.
* Virtual Machine SCSI Controller Performance Object
  + The Virtual Machine SCSI Controller performance object provides SCSI controller performance counters for a virtual machine.
* Virtual Machine I/O APIC Device Performance Object
  + The Virtual Machine I/O APIC Device performance object provides I/O APIC device performance counters for a virtual machine.
* Virtual Machine Local APIC Performance Object
  + The Virtual Machine Local APIC performance object provides local APIC performance counters for a virtual machine.
* Virtual Machine Programmable Interrupt Controller (PIC) Performance Object
  + The Virtual Machine PIC performance object provides PIC performance counters for a virtual machine.
* Virtual Machine Remoting System Performance Object
  + The Virtual Machine remoting system performance object provides remoting system performance counters for a virtual machine.
* Virtual Machine Video System Performance Object
  + The Virtual Machine video system performance object provides video system performance counters for a virtual machine.
* Virtual Machine Guest Memory Operations: Save, Snapshot and Restore System Performance Object
  + The Virtual Machine Guest Memory Operations: Save, Snapshot and Restore system performance object provides Guest Memory Operations: Save, Snapshot and Restore system performance counters for a virtual machine.
* Virtual Machine Global information about Hypervisor Performance Object
  + The Virtual Machine Global information about Hypervisor performance object provides Global information about Hypervisor performance counters for a virtual machine such as the number of logical processors in the system and the number of virtual machines present in the system.
* Virtual Machine Information about logical processors Performance Object
  + The Virtual Machine Information about logical processors performance object provides Information about logical processors performance counters for a virtual machine such as the total number of hardware interrupts received by a logical processor and the total number of virtual processor context switches on a logical processor.
* Virtual Machine Information about Partition Performance Object
  + The Virtual Machine Information about Partition performance object provides Information about Partition performance counters for a virtual machine such as the number of existing virtual processors, the total number of page tables, and more.
* Virtual Machine Information about Virtual Processor Performance Object
  + The Virtual Machine Information about Virtual Processor performance object provides Information about Virtual Processor performance counters for a virtual machine such as the total number of intercepts that the hypervisor received while executing guest code on a virtual processor, the cycles that a virtual processor had spent handling all intercepts, and more.
* Virtual Machine Virtual Infrastructure Driver (VID) Partition Performance Object
  + The Virtual Machine VID partition performance object provides VID partition performance counters for a virtual machine such as the total number of registered handlers, the total number of registered IO ports, and more.
* Virtual Machine VID Message Queue Performance Object
  + The Virtual Machine VID message queue performance object provides VID message queue performance counters for a virtual machine such as the total number of registered handlers, the total number of registered IO ports, and more.
* Virtual Machine Worker Process Memory Manager Performance Object
  + The Virtual Machine Worker Process Memory Manager performance object provides worker process memory manager performance counters for a virtual machine such as the total number of memory block count, the total number of RAM memory block count, and more.
* Virtual Machine Virtual Network Adapter Performance Object
  + The Virtual Machine Virtual Network Adapter performance object provides virtual network adapter performance counters for a virtual machine such as the total number of bytes that have traversed the network adapter, the total number of bytes per second traversing the network adapter, and more.
* Virtual Machine Virtualization Switch Port Performance Object
  + The Virtual Machine Virtualization Switch Port performance object provides virtualization switch port performance counters for a virtual machine such as the total number of bytes that have traversed the virtual switch port, the total number of bytes per second traversing the virtual switch port, and more.
* Virtual Machine Virtual Storage Device Performance Object
  + The Virtual Machine Virtual Storage Device performance object provides virtual storage device performance counters for a virtual machine such as the total number of read operations that have occurred on the virtual storage device, the total number of write operations that have occurred on the virtual storage device, and more.
* Virtual Machine Virtual CDROM Device Performance Object
  + The Virtual Machine Virtual CDROM Device performance object provides virtual CDROM device performance counters for a virtual machine such as the total number of read operations that have occurred on the virtual CDROM device, the total number of write operations that have occurred on the virtual CDROM device, and more.
* Virtual Machine Storage Virtualization service provider (VSP) Performance Object
  + The Virtual Machine Storage VSP performance object provides storage VSP performance counters for a virtual machine such as the total number of read operations that have occurred on the storage VSP, the total number of write operations that have occurred on the storage VSP, and more.
* Virtual Machine Network Virtualization service provider (VSP) Performance Object
  + The Virtual Machine Network VSP performance object provides network VSP performance counters for a virtual machine such as the total number of read operations that have occurred on the network VSP, the total number of write operations that have occurred on the network VSP, and more.
* Virtual Machine Virtual Machine Bus Performance Object
  + The Virtual Machine Virtual Machine Bus performance object provides virtual machine bus performance counters for a virtual machine such as the total number of interrupts received by the virtual machine bus, the total number of interrupts sent by the virtual machine bus, and more.
* Virtual Machine Management Service Virtual Machine Summary Performance Object
  + The Virtual Machine Management Service Virtual Machine Summary performance object provides virtual machine performance summary counters such as the number of virtual machine’s that are running, the number of virtual machine’s that are turned off, and more.
* Virtual Machine Management Service WMI Event Provider Performance Object
  + The Virtual Machine Management Service WMI Event Provider performance object provides virtual machine management service WMI event provider performance counters such as the number of WMI \_\_InstanceCreationEvent indications, the number of WMI \_\_InstanceModificationEvent indications, and more.
* Virtual Machine Management Service WMI Object Provider Summary Performance Object
  + The Virtual Machine Management Service WMI Object Provider Summary performance object provides virtual machine management service WMI object provider performance summary counters such as the number of WMI requests that are in progress, the number of WMI requests processed, and more.
* Virtual Machine Management Service WMI Object Provider Detail Performance Object
  + The Virtual Machine Management Service WMI Object Provider Detail performance object provides virtual machine management service WMI object provider performance detail counters such as the number of CreateInstanceEnum WMI requests that are in progress, the number of CreateInstanceEnum WMI requests processed, and more.
* Virtual Machine Management Service Task Manager Summary Performance Object
  + The Virtual Machine Management Service Task Manager Summary performance object provides virtual machine management service task manager performance summary counters such as the number of tasks that are in progress, the number of tasks completed, and more.
* Virtual Machine Management Service Task Manager Detail Performance Object
  + The Virtual Machine Management Service Task Manager Detail performance object provides virtual machine management service task manager performance detail counters such as the average time to execute a task in 100ns units, the minimum time to execute a task in 100ns units, and more.
* Windows OS [TPM Base Services (TBS)](http://msdn.microsoft.com/en-us/library/aa446792(VS.85).aspx) Performance Object
  + The Windows OS TBS performance object provides the Windows OS TBS performance counters such as the number of TBS contexts that are currently active and the number of resources that are currently being managed by the Windows OS TBS.
* Windows OS RPC/HTTP Proxy Performance Object
  + The Windows OS RPC/HTTP Proxy performance object provides the Windows OS RPC/HTTP Proxy performance counters such as the number of unique users currently connected to a back-end server via RPC/HTTP, the rate at which RPC/HTTP Proxy attempts to establish a connection to a back-end server, and more.
* Windows OS [Netlogon](http://download.microsoft.com/download/9/5/E/95EF66AF-9026-4BB0-A41D-A4F81802D92C/%5BMS-NRPC%5D.pdf) Performance Object
  + The Windows Netlogon performance object provides the Windows Netlogon performance counters such as the number of times the semaphore has been acquired over the lifetime of the Secure Channel connection, the number of times a thread has timed out waiting for the semaphore over the lifetime of the Secure Channel connection, and more.
* Windows OS [Authorization Manager Applications](http://technet.microsoft.com/en-us/library/cc732290.aspx) Performance Object
  + The Windows OS [Authorization Manager Applications](http://technet.microsoft.com/en-us/library/cc732290.aspx) performance object provides the Windows OS Authorization Manager enabled applications performance counters such as the number of authorization scopes in an application, and the number of authorization scopes that are currently loaded in memory.
* Windows OS [HTTP Service](http://msdn.microsoft.com/en-us/library/aa364510.aspx) General Performance Object
  + The Windows OS HTTP Service general performance object provides the Windows HTTP Service general performance counters such as the number of URIs currently cached by the kernel http.sys, the number of URIs that have been removed from the kernel URI cache, and more.
* Windows OS [HTTP Service](http://msdn.microsoft.com/en-us/library/aa364510.aspx) URL Group-specific Performance Object
  + The Windows OS HTTP Service URL group-specific performance object provides the Windows HTTP Service URL group-specific performance counters such as the rate of sending data by HTTP service for a web site, the rate of received data by HTTP service for a web site, and more.
* Windows OS [HTTP Service](http://msdn.microsoft.com/en-us/library/aa364510.aspx) Request Queues Performance Object
  + The Windows OS HTTP Service Request Queues performance object provides the Windows HTTP Service request queue performance counters such as the number of requests rejected from a queue, the rate at which requests are arriving in the queue, and more.
* Windows OS [Windows Filtering Platform](http://msdn.microsoft.com/en-us/library/aa366510(VS.85).aspx) IPV4 Traffic and Connection Performance Object
  + The Windows OS Windows Filtering Platform IPV4 Traffic and Connection performance object provides the Windows Filtering Platform IPV4 traffic and connection performance counters such as the rate at which inbound packets are discarded by the Windows Filtering Platform, the rate at which outbound packets are discarded by the Windows Filtering Platform, and more.
* Windows OS [Windows Filtering Platform](http://msdn.microsoft.com/en-us/library/aa366510(VS.85).aspx) IPV6 Traffic and Connection Performance Object
  + The Windows OS Windows Filtering Platform IPV6 Traffic and Connection performance object provides the Windows Filtering Platform IPV6 traffic and connection performance counters such as the rate at which inbound packets are discarded by the Windows Filtering Platform, the rate at which outbound packets are discarded by the Windows Filtering Platform, and more.
* Windows OS [Windows Filtering Platform](http://msdn.microsoft.com/en-us/library/aa366510(VS.85).aspx) Provider Registration Performance Object
  + The Windows OS Windows Filtering Platform Provider Registration performance object provides the Windows Filtering Platform provider registration performance counters such as the number of providers registered with the Windows Filtering Platform.
* Windows OS [Windows Filtering Platform](http://msdn.microsoft.com/en-us/library/aa366510(VS.85).aspx) Internet Protocol security (IPsec) general Performance Object
  + The Windows OS Windows Filtering Platform IPsec general performance object provides the Windows Filtering Platform IPsec general performance counters such as the number of active Quick Mode security associations, the rate of bytes received using Tunnel Mode, and more.
* Windows OS [Windows Filtering Platform](http://msdn.microsoft.com/en-us/library/aa366510(VS.85).aspx) IPsec IPV4 Internet Key Exchange (IKE) Performance Object
  + The Windows OS Windows Filtering Platform IPsec IPV4 IKE performance object provides the Windows Filtering Platform IPsec IPV4 IKE performance counters such as the number of currently active Main Mode security associations, the number of Main Mode negotiations attempted since IPsec was last started, and more.
* Windows OS [Windows Filtering Platform](http://msdn.microsoft.com/en-us/library/aa366510(VS.85).aspx) IPsec IPV6 Internet Key Exchange (IKE) Performance Object
  + The Windows OS Windows Filtering Platform IPsec IPV6 IKE performance object provides the Windows Filtering Platform IPsec IPV6 IKE performance counters such as the number of currently active Main Mode security associations, the number of Main Mode negotiations attempted since IPsec was last started, and more.
* Windows OS [Windows Filtering Platform](http://msdn.microsoft.com/en-us/library/aa366510(VS.85).aspx) IPsec IPV4 Authenticated IP (AuthIP) Performance Object
  + The Windows OS Windows Filtering Platform IPsec IPV4 AuthIP performance object provides the Windows Filtering Platform IPsec IPV4 AuthIP performance counters such as the number of pending Main Mode negotiations, the rate at which Main Mode negotiations are being attempted, and more.
* Windows OS [Windows Filtering Platform](http://msdn.microsoft.com/en-us/library/aa366510(VS.85).aspx) IPsec IPV6 Authenticated IP (AuthIP) Performance Object
  + The Windows OS Windows Filtering Platform IPsec IPV6 AuthIP performance object provides the Windows Filtering Platform IPsec IPV6 AuthIP performance counters such as the number of pending Main Mode negotiations, the rate at which Main Mode negotiations are being attempted, and more.
* Windows OS [Windows Filtering Platform](http://msdn.microsoft.com/en-us/library/aa366510(VS.85).aspx) IPsec Internet Key Exchange (IKE) and Authenticated IP (AuthIP) general Performance Object
  + The Windows OS Windows Filtering Platform IPsec IKE and AuthIP general performance object provides the Windows Filtering Platform IPsec IKE and AuthIP general performance counters such as the number of milliseconds taken for the last IKE Main Mode security association negotiated, the number of milliseconds taken for the last Authenticated IP Main Mode security association negotiated, and more.
* Windows OS Network Policy Server (NPS) [Host Credential Authorization Protocol (HCAP)](http://technet.microsoft.com/en-us/library/cc732681.aspx) Performance Object
  + The Windows OS NPS HCAP performance object provides the NPS HCAP performance counters such as the average time taken by the NPS to validate HCAP requests in milliseconds for this uptime, the number of successful [Posture Validation](http://technet.microsoft.com/en-us/library/dd314201.aspx) responses sent for this uptime, and more.
* Windows OS Graphics Adapter Direct Memory Access (DMA) Buffering Performance Object
  + The Windows OS Graphics Adapter DMA Buffering performance object provides the graphics adapter DMA buffering performance counters such as the number of DMA buffer references attempted, the number of allocations referenced with valid prepatching, and more.
* Windows OS Graphics Adapter Allocation Performance Object
  + The Windows OS Graphics Adapter Allocation performance object provides the graphics adapter allocation performance counters such as the number of allocations created/sec, the number of allocations destroyed/sec, and more.
* Windows OS Graphics Adapter Swizzling and Renaming Performance Object
  + The Windows OS Graphics Adapter Swizzling and Renaming performance object provides the graphics adapter swizzling and renaming performance counters such as the number of swizzling range acquired/sec, the number of swizzling range released/sec, and more.
* Windows OS Graphics Adapter Lock Performance Object
  + The Windows OS Graphics Adapter Lock performance object provides the graphics adapter lock performance counters such as the number of locks/sec, the number of locks with no-overwrite per second, and more.
* Windows OS Graphics Adapter Preparation Performance Object
  + The Windows OS Graphics Adapter Preparation performance object provides the graphics adapter preparation performance counters such as the number of times preparation thread stalled because an engine had not finished with DMA buffer, the number of DMA buffer prepared/sec, and more.
* Windows OS Graphics Adapter Fault Performance Object
  + The Windows OS Graphics Adapter Fault performance object provides the graphics adapter fault performance counters such as the number of evictions caused by purge, the number of evictions caused by suspended CPU access, and more.
* Windows OS Graphics Adapter Memory Transfer Performance Object
  + The Windows OS Graphics Adapter Memory Transfer performance object provides the graphics adapter memory transfer performance counters such as the number of bytes filled in memory/sec, the number of bytes mapped into aperture per second, and more.
* Windows OS [Background Intelligent Transfer Service (BITS)](http://download.microsoft.com/download/f/5/5/f557fec8-5096-4175-b099-18e5f28f11ad/Changes_in_Functionality_from_Windows_Server_2003_with_SP1_to_Windows_Server_2008.doc) Network Performance Object
  + The Windows OS BITS Network performance object provides the BITS network performance counters such as the estimated bandwidth available to the remote system (Bits/sec), the BITS download response interval (msec), and more.
* Windows OS [Certificate Server](http://technet.microsoft.com/en-us/library/cc770357.aspx) Performance Object
  + The Windows OS Certificate Server performance object provides the Certificate Server performance counters such as the number of certificate requests processed per second, the number of issued certificate requests processed per second, and more.
* Windows OS [Certificate Server](http://technet.microsoft.com/en-us/library/cc770357.aspx) Connection Performance Object
  + The Windows OS Certificate Server Connection performance object provides the Certificate Server connection performance counters such as the number of active connections.
* Windows OS [Certificate Server](http://technet.microsoft.com/en-us/library/cc770357.aspx) Online Certificate Status Protocol (OCSP) Processing Performance Object
  + The Windows OS Certificate Server OCSP Processing performance object provides the Certificate Server OCSP processing performance counters such as the number of requests processed per second, the number of failed requests processed per second, and more.
* Windows OS [Certificate Server](http://technet.microsoft.com/en-us/library/cc770357.aspx) OCSP Connection Performance Object
  + The Windows OS Certificate Server OCSP Connection performance object provides the Certificate Server OCSP connection performance counters such as the number of active OCSP connections.
* Windows OS [Certificate Server](http://technet.microsoft.com/en-us/library/cc770357.aspx) OCSP ISAPI based Web Application Performance Object
  + The Windows OS Certificate Server OCSP ISAPI based Web Application performance object provides the Certificate Server OCSP ISAPI based web application performance counters such as the average time spent handling a request, the number of current requests waiting to be handled, and more.

## Addressing 5.2.1.3 “The OS shall detect and send out reports to registered administrators when the OS service level of system resources is lower than the predetermined minimum”

The Windows OS addresses the “5.2.1.3” requirement. As described in the “Windows OS Reliability and Performance Monitor” section under Appendix B of this paper, the following specific performance objects provide corresponding performance counters for monitoring system resources.

* “[Memory Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)”;
* “[Objects Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)”;
* “[Paging File Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)”;
* “[Physical Disk Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)”;
* “[Process Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)”;
* “[Process Address Space Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)”;
* “[Processor Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)”;
* “[System Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)”;
* “[Thread Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)”;
* “[Thread Details Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)”;
* “[WSRM Process Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)”;
* “[WSRM Process Matching Criteria Performance Object](http://technet.microsoft.com/en-us/library/cc783501.aspx)”.

To achieving an alert for sending out reports after detecting that the threshold value for a specific counter is reached or fallen below, a [Windows Management Instrumentation (WMI) task](http://msdn.microsoft.com/en-us/library/aa394601(VS.85).aspx) can be defined for the counter belonging to a [Performance Data Collector Set](http://technet.microsoft.com/en-us/library/cc749267.aspx)”.

## Addressing 5.2.1.4 “The OS shall adopt appropriate policies for providing the priority to the subject for the use of a resource subset within the OS based on the limited service priority”

The Windows OS addresses the “5.2.1.4” requirement because of the Windows OS [priority scheduling scheme](http://msdn.microsoft.com/en-us/library/ms685100(VS.85).aspx) implemented by the Windows OS process manager (ps.lib), as described in the following “Windows OS Priority Scheduling Scheme” section.

### Windows OS Priority Scheduling Scheme

The Windows OS [priority scheduling scheme](http://msdn.microsoft.com/en-us/library/ms685100(VS.85).aspx) is implemented by the Windows OS process manager (ps.lib). It possesses the following characteristics.

Windows OS threads are scheduled to run based on their scheduling priority. Each thread is assigned a scheduling priority. The priority levels range from zero (lowest priority) to 31 (highest priority).

The Windows OS treats all threads with the same priority as equal. The Windows OS assigns time slices in a round-robin fashion to all threads with the highest priority. If none of these threads are ready to run, the Windows OS assigns time slices in a round-robin fashion to all threads with the next highest priority. If a higher-priority thread becomes available to run, the Windows OS ceases to execute the lower-priority thread (without allowing it to finish using its time slice), and assigns a full time slice to the higher-priority thread after the necessary context switching.

The priority of each Windows OS thread is determined by the following criteria:

* The priority class of its Windows OS process;
* The priority level of the thread within the priority class of its Windows OS process;
* The priority class and priority level are combined to form the base priority of a Windows OS thread.

By default, the priority class of a Windows OS process is NORMAL\_PRIORITY\_CLASS. The [CreateProcess()](http://msdn.microsoft.com/en-us/library/ms682425.aspx) function is used to specify the priority class of a child process as the child process is created. If the caller process belongs to IDLE\_PRIORITY\_CLASS or BELOW\_NORMAL\_PRIORITY\_CLASS, the new child process inherits this class also. The [GetPriorityClass()](http://msdn.microsoft.com/en-us/library/ms683211(VS.85).aspx) function is used to determine the current priority class of a running process and the [SetPriorityClass()](http://msdn.microsoft.com/en-us/library/ms686219(VS.85).aspx) function is used to change the priority class of a running process. Increasing the base priority of a target process requires the caller subject to possess the [SeIncreaseBasePriorityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) and the [PROCESS\_SET\_INFORMATION](http://msdn.microsoft.com/en-us/library/ms684880.aspx) right to the target process. By default, the [SeIncreaseBasePriorityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) is assigned only to local system and the “Administrators” group.

All Windows OS threads are created using THREAD\_PRIORITY\_NORMAL. This means that the thread priority is the same as the process priority class. After a thread is created, the [SetThreadPriority()](http://msdn.microsoft.com/en-us/library/ms686277(VS.85).aspx) function is used to adjust its priority relative to other threads in the same Windows OS process. Increasing the base priority of a target thread requires the caller subject to possess the [SeIncreaseBasePriorityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) and the [THREAD\_SET\_LIMITED\_INFORMATION](http://msdn.microsoft.com/en-us/library/ms686769(VS.85).aspx) right to the target thread.

Each Windows OS thread has a dynamic priority. This is the priority the Windows OS scheduler uses to determine which thread to execute. Initially, a thread’s dynamic priority is the same as its base priority. The Windows OS can boost and lower the dynamic priority, to ensure that it is responsive and that no threads are starved for processor time. The Windows OS does not boost the priority of threads with a base priority level between 16 and 31. Only threads with a base priority between 0 and 15 receive dynamic priority boosts.

When a Windows OS process that uses NORMAL\_PRIORITY\_CLASS is brought to the foreground, the Windows OS scheduler boosts the priority class of the process associated with the foreground window, so that it is greater than or equal to the priority class of any background processes. The priority class returns to its original setting when the process is no longer in the foreground.

When a window receives input, such as timer messages, mouse messages, or keyboard input, the Windows OS scheduler boosts the priority of the thread that owns the window.

When the wait conditions for a blocked thread are satisfied, the scheduler boosts the priority of the thread. For example, when a wait operation associated with disk or keyboard I/O finishes, the thread receives a priority boost.

The priority-boosting feature can be disabled and enabled by calling the [SetProcessPriorityBoost()](http://msdn.microsoft.com/en-us/library/ms686225.aspx) or [SetThreadPriorityBoost()](http://msdn.microsoft.com/en-us/library/ms686280(VS.85).aspx) function. It is enabled by default. To disable or enable the priority-boosting feature, possessing the [PROCESS\_SET\_INFORMATION](http://msdn.microsoft.com/en-us/library/ms684880.aspx) right to the target process or the [THREAD\_SET\_LIMITED\_INFORMATION](http://msdn.microsoft.com/en-us/library/ms686769(VS.85).aspx) right to the target thread respectively is necessary.

After raising a Windows OS thread’s dynamic priority, the Windows OS scheduler reduces that priority by one level each time the thread completes a time slice, until the thread drops back to its base priority. A thread’s dynamic priority is never less than its base priority.

## Addressing 5.2.1.5 “The OS shall implement the management and allocation of the OS resources in accordance with requirements of ceiling on resources allocation, and make sure that users/subjects will not monopolize any managed resource by implementing the quota system”

For the storage quota aspect, the “5.2.1.5” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “5.2.1.1” requirement “The OS shall enforce maximum quotas on the portion of shared persistent storage that individual authorized users can use”.

For the storage quota aspect, the Windows OS also addresses the “5.2.1.5” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “5.2.1.1” requirement for detail.

The Windows OS additionally addresses the “5.2.1.5” requirement for the below quota aspects on a per user account basis, as described in the following “Windows OS Quota Control for CPU Rate and Memory Usage” section.

* Paged pool memory;
* Non paged pool memory;
* Paging file;
* Working set pages;
* CPU time rate.

### Windows OS Quota Control for CPU Rate and Memory Usage

In the Windows OS, a quota facility is available in every Windows OS process for controlling the CPU rate and the memory consumption. The Windows OS process manager (ps.lib) is the actual maintainer of this quota facility. This quota facility addresses the following CPU and memory aspects on a per user account basis.

* Paged pool memory
  + The paged pool is an area of system physical memory that can be consumed by a subject through the Windows OS mediation for the allocation of objects which can be written to disk when they are not being used.
  + By default, there is no limit, and the Windows OS expands automatically when needed.
  + A value in bytes can be configured for a user account as the quota limit for this “paged pool memory” quota aspect.
  + The status code [STATUS\_QUOTA\_EXCEEDED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) is resulted when the quota limit is exceeded.
  + While the Windows OS process manager (ps.lib) is the maintainer of this quota aspect, the Windows OS memory manager (mm.lib) is the actual enforcer.
* Non paged pool memory
  + The non paged pool is an area of system physical memory that can be consumed by a subject through the Windows OS mediation for the allocation of objects which are not written to disk.
  + By default, there is no limit, and the Windows OS expands automatically when needed.
  + A value in bytes can be configured for a user account as the quota limit for this “non paged pool memory” quota aspect.
  + The status code [STATUS\_QUOTA\_EXCEEDED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) is resulted when the quota limit is exceeded.
  + While the Windows OS process manager (ps.lib) is the maintainer of this quota aspect, the Windows OS memory manager (mm.lib) is the actual enforcer.
* Paging file
  + The paging file is a reserved space on disk (in the form of a physical file) that backs up committed physical memory on the computer.
  + By default, there is no limit.
  + A value in bytes can be configured for a user account as the quota limit for this “paging file” quota aspect.
  + The status code [STATUS\_PAGEFILE\_QUOTA\_EXCEEDED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) is resulted when the quota limit is exceeded.
  + While the Windows OS process manager (ps.lib) is the maintainer of this quota aspect, the Windows OS memory manager (mm.lib) is the actual enforcer.
* Working set pages
  + The working set pages of a Windows OS process is the set of memory pages that is visible to the process in the physical memory. These pages are available to the process without triggering a page fault.
  + By default, there is no limit.
  + A value in number of pages can be configured for a user account as the quota limit for this “working set pages” quota aspect.
  + The status code [STATUS\_WORKING\_SET\_QUOTA](http://msdn.microsoft.com/en-us/library/cc704588.aspx) is resulted when the quota limit is exceeded.
  + While the Windows OS process manager (ps.lib) is the maintainer of this quota aspect, the Windows OS memory manager (mm.lib) is the actual enforcer.
* CPU time rate
  + The CPU time rate is the rate at which the CPU time is being consumed.
  + By default, the rate is 100 percent.
  + A value in percentage can be configured for a user account as the quota limit for this “CPU time rate” quota aspect.
  + While no specific status code is resulted when the quota limit is exceeded, the thread that has triggered the quota limit excess is swapped out from the CPU execution through the thread switching mechanism.
  + While the Windows OS process manager (ps.lib) is the maintainer of this quota aspect, the Windows OS microkernel component is the actual enforcer.

To configure a specific (non default) quota value of one of the above CPU and memory aspects for a user account, an authorized administrator enters the desire value as registry key value data in the form of a 32-bit number (REG\_DWORD) under the user SID registry subkey (representing the target user account) residing in the following registry key.

* HKLM\System\CurrentControlSet\Control\Session Manager\Quota System
  + By default, the write access to this registry key is limited to local system and the “Administrators” group.

The specific registry key values under the target user SID registry subkey are:

* “PagedPoolQuota”
  + This registry key value name corresponds to the “paged pool memory” quota aspect;
* “NonPagedPoolQuota”
  + This registry key value name corresponds to the “non paged pool memory” quota aspect;
* “PagingFileQuota”
  + This registry key value name corresponds to the “paging file” quota aspect;
* “WorkingSetPagesQuota”
  + This registry key value name corresponds to the “working set pages” quota aspect;
* “CpuRateLimit”
  + This registry key value name corresponds to the “CPU time rate” quota aspect.

## Addressing 5.2.1.6 “The OS shall ensure resources can be accessed and utilized when an authorized subject send out request”

The Windows OS addresses the “5.2.1.6” requirement. Due to the availability of a quota system for the following aspects, the possibility where an authorized subject is unable to gain access to the following resources due to resource exhaustion is limited.

* Disk for persistent storage;
* Paged pool memory;
* Non paged pool memory;
* Paging file;
* Working set pages;
* CPU time rate.

## Addressing 5.2.1.7 “The OS shall provide a mechanism based on one user or one user group to and control the consumption of disk”

The “5.2.1.7” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “5.2.1.1” requirement “The OS shall enforce maximum quotas on the portion of shared persistent storage that individual authorized users can use”.

The Windows OS also addresses the “5.2.1.7” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “5.2.1.1” requirement for detail.

## Addressing 5.2.1.8 “The OS shall provide a mechanism based on one user or one user group to and control the utilization of CPU”

The Windows OS addresses the “5.2.1.8” requirement for the following quota aspect on a per user account basis, as described in the in the “Windows OS Quota Control for CPU Rate and Memory Usage” section under Appendix B of this paper.

* CPU time rate.

# Meeting Additional “Information Protection Resource Control Management Requirements”

The Commercial Grade OS Requirement Set already has 1 individual management requirement under the heading of “Information Protection Resource Control Management Requirements” and it is the “5.2.2.1” requirement.

There is 1 additional individual management requirement under the heading of “Information Protection Resource Control Management Requirements” in this appendix. It is listed as “5.2.2.2”.

## Addressing 5.2.2.2 “The OS shall implement the management and allocation of the OS resources”

For the storage quota aspect, the “5.2.2.2” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “5.2.2.1” requirement “The OS shall allow only authorized administrators the ability to set maximum quotas on shared persistent storage”.

For the storage quota aspect, the Windows OS also addresses the “5.2.2.2” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “5.2.2.1” requirement for detail.

The Windows OS additionally addresses the “5.2.2.2” requirement for the following quota aspects on a per user account basis, as described in the “Windows OS Quota Control for CPU Rate and Memory Usage” section under Appendix B of this paper. Specifically, only an administrator is allowed to configure the quota for the following aspects:

* Paged pool memory;
* Non paged pool memory;
* Paging file;
* Working set pages;
* CPU time rate.

# Meeting Additional “Information Protection Resource Control Audit Requirements”

The Commercial Grade OS Requirement Set already has 2 individual audit requirements under the heading of “Information Protection Resource Control Audit Requirements” and they are the “5.2.3.1” and “5.2.3.2” requirements.

There are 2 additional individual management requirements under the heading of “Information Protection Resource Control Audit Requirements” in this appendix. They are listed as “5.2.3.n”, where n = 3 and 4.

## Addressing 5.2.3.3 “The OS shall point out the least types of recordable audit events, including the operations of system administrator and system security administrator and others”

The “5.2.3.3” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “5.2.3.2” requirement “The OS shall provide the ability to audit the setting of maximum quotas on shared persistent storage”.

The Windows OS equally does not meet the “5.2.3.3” requirement for the auditing of the disk storage quota configuration aspect. We recommend (as a workaround for the Commercial Grade OS Requirement Set “5.2.3.2” requirement) that an organization monitors the setting of the user quota information entry for a specific user account and the setting of the default quota information entry of specific targeted NTFS non read-only volumes for changes. Please see the justification text for addressing the Commercial Grade OS Requirement Set “5.2.3.2” requirement for detail.

However, the Windows OS meets the “5.2.3.3” requirement for the auditing of the CPU rate and memory usage quota configuration aspects. The generation of [Event ID 4657](http://support.microsoft.com/kb/947226) “A registry value was modified” (SE\_AUDITID\_ETW\_REG\_VALUE\_CHANGE\_AUDIT\_value) security audit record for success captures the auditing of the CPU rate and memory usage quota configuration aspects because the CPU rate and memory usage quota configuration occurs within the following registry key:

* HKLM\System\CurrentControlSet\Control\Session Manager\Quota System.

Please see the “Windows OS Quota Control for CPU Rate and Memory Usage” section under Appendix B of this paper for detail.

## Addressing 5.2.3.4 “The OS shall point out the least types of recordable audit events, including the deletion of objects”

The Windows OS addresses the “5.2.3.4” requirement as the Windows OS generates an [Event ID 4660](http://support.microsoft.com/kb/947226) “An object was deleted” (SE\_AUDITID\_ETW\_DELETE\_OBJECT\_value) security audit record for success with the following informational items when available:

* Security ID:
* Account Name:
* Account Domain:
* Logon ID:
* Object Server:
* Handle ID:
* Process ID:
* Process Name:
* Transaction ID:.

# Meeting Additional “Information Protection Self Testing Functional Requirements”

The Commercial Grade OS Requirement Set already has 2 removed individual functional requirements under the heading of “Information Protection Self Testing Functional Requirements”. There are listed as “5.3.1.1” and “5.3.1.2”. The Commercial Grade OS Requirement Set also has 1 individual functional requirement under the heading of “Information Protection Trusted Initialization Functional Requirements”. It is listed as “5.3.1.3”.

There is 1 additional individual management requirements under the heading of “Information Protection Self Testing Functional Requirements” in this appendix. It is listed as “5.3.1.4”.

## Addressing 5.3.1.4 “The OS shall provide the mechanisms and processes for regular confirmation of correct OS operations”

The “5.3.1.4” requirement additionally requires that “these mechanisms and processes shall involve the supervision of system resources, correct operations of hardware and firmware, and inspection of error states that may be transmitted within the entire OS, as well as the inspection of communication error beyond the threshold regulated by users, etc”

The “5.3.1.4” requirement treats similar security concern(s) as the removed Commercial Grade OS Requirement Set “5.3.1.1” requirement “The OS shall run a suite of self tests to demonstrate the correct operation of the OS security functions during specific occasions”. The removed Commercial Grade OS Requirement Set “5.3.1.1” requirement requires the following specific occasions to run the self tests.

1. The initial start-up;
2. At the request of an authorized administrator;
3. Periodically (at an authorized administrator defined frequency) during normal operations.

The Windows OS equally does not meet the “5.3.1.4” requirement as the Windows OS does not incorporate self tests for its security functions in order to achieve regular confirmation of correct system (security) operation.

However, as parts of its serviceability and sustained engineering processes, Microsoft continues to conduct unit tests and scenario based tests for the Windows OS after the Windows OS is shipped to customers. Issues arisen during the serviceability and sustained engineering processes are investigated and addressed in the next scheduled service pack of the Windows OS.

# Meeting the “Information Protection Persistent Storage Data Availability Functional Requirements”

There are 14 individual functional requirements under the heading of “Information Protection Persistent Storage Data Availability Functional Requirements”. They are listed as “5.4.1.n”, where n = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, and 14.

## Addressing 5.4.1.1 “While handling data access, the OS shall check whether requested data on storage medium has corruption errors, and restore the data if a corruption error is detected”

The Windows OS addresses the “5.4.1.1” requirement because the default capability of the [self healing NTFS](http://download.microsoft.com/download/8/1/E/81EFEE63-F192-4BD7-AE85-2DDBDA888EEE/WS2008RAS.docx), as explained in the following “Self healing Windows OS NTFS” section.

### Self healing Windows OS NTFS

Disk corruptions can happen for any number of reasons; a few of the more prominent ones are IO errors due to improper shutdown, physical disk surface degradation and software bugs. This type of error can leave the file system in an inconsistent state that might prevent access to the data until some form of corrective action is taken to get it back to a consistent state. The [self healing NTFS](http://download.microsoft.com/download/8/1/E/81EFEE63-F192-4BD7-AE85-2DDBDA888EEE/WS2008RAS.docx) is a set of enhancements to the NTFS file system code base to support the real-time correction of detected corruption to the on-disk meta-data.

When the Windows OS NTFS handles an authorized subject’s request to access a file residing in a NTFS partition/volume, it reads and processes elements of meta-data (such as the [Master File Table (MFT)](http://msdn.microsoft.com/en-us/library/bb470206(VS.85).aspx)) associated with the file of the specific NTFS partition/volume. It verifies the meta-data elements as they are read and processed from the disk. Without the NTFS self-healing, when NTFS detects meta-data corruption, it raises an error status ([STATUS\_DISK\_CORRUPT\_ERROR](http://msdn.microsoft.com/en-us/library/cc704588.aspx)) indicating a corruption. With the NTFS self-healing at the point where a corruption is detected, NTFS determines the scope of the repair needed if the specific NTFS partition/volume is not write-protected or dismounted. If the detected corruption is in the list of corruption types it can handle, NTFS takes appropriate corrective action. Such disk correction happens in the background and does not affect the servers operation. If the detected corruption is not of a type that can be handled automatically, the specific NTFS partition/volume is marked as dirty and the traditional [chkdsk command line tool](http://technet.microsoft.com/en-us/library/cc730714.aspx) will be required to be run.

The Windows OS NTFS generates one of the following two soft audit event records to the soft audit storage[[36]](#footnote-37)

* Event ID 130 (NTFS\_ETW\_REPAIR\_SUCCESS) “The file system structure on volume <Volume ID> has now been repaired”;
* Event ID 131 (NTFS\_ETW\_REPAIR\_FAILED) “The file system structure on volume <Volume ID> cannot be corrected. Please run the chkdsk utility on the volume <Volume ID>”,

after the Windows OS NTFS concludes the attempt of a self-healing repair run. We note that the chkdsk utility mentioned in Event ID 131 is the [chkdsk command line tool](http://technet.microsoft.com/en-us/library/cc730714.aspx) available to the administrators.

In either Event ID 130 or Event 131 event record, there is a “RepairDetail” field which contains the detail about a specific detected corruption that is or is not fixed respectively in the specific self-healing repair run.

The following may be included in the “RepairDetail” field.

* The following is a report of the start of a specific self healing repair run.
  + MSG SHLOG (self healing log message) REPAIR\_START\_TIME
    - “Start repair on <the current time when the specific self-healing repair run starts>”.
* The following is a report of the end of a specific self healing repair run.
  + MSG SHLOG REPAIR\_END\_TIME
    - “End repair on <the current time when the specific self-healing repair run ends>”.
* The following are reports of specific corruptions being detected (appearing between the start report and the end report of a specific self healing repair run).
  + MSG SHLOG UNNAMED\_INDEX\_ENTRY\_POINTS\_TO\_FREE\_FRS
    - “An index entry of index <ParentFileReference of the parent directory> points to unused file <FileReference of the file/directory being opened>”.
  + MSG SHLOG INCORRECT\_FRS\_MULTI\_SECTOR\_HEADER\_SIGNATURE
    - “The multi-sector header signature in file <FileReference of the file/directory being opened> is incorrect”.
  + MSG SHLOG FRS\_USA\_OFFSET\_BELOW\_MINIMUM
    - “The [Update Sequence Array (USA) offset](http://msdn.microsoft.com/en-us/library/bb470212(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened> is too small”.
  + MSG SHLOG INCORRECT\_FRS\_USA\_OFFSET
    - “The [Update Sequence Array (USA) offset](http://msdn.microsoft.com/en-us/library/bb470212(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened> is incorrect”.
  + MSG SHLOG INCORRECT\_FRS\_USA\_SIZE
    - “The [Update Sequence Array (USA) size](http://msdn.microsoft.com/en-us/library/bb470212(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened> is incorrect”.
  + MSG SHLOG INCORRECT\_FIRST\_ATTR\_OFFSET
    - “The [first attribute offset](http://msdn.microsoft.com/en-us/library/bb470124(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened> is incorrect”.
  + MSG SHLOG INCORRECT\_FRS\_HEADER
    - “The [bytes available](http://msdn.microsoft.com/en-us/library/bb470124(VS.85).aspx), <the current value>, in the file record segment header for file <FileReference of the file/directory being opened> is incorrect”.
  + MSG SHLOG OBJID\_INDEX\_ENTRY\_WITH\_NO\_OBJID\_FRS
    - “The object id index entry in file <FileReference of the object ID index table file> points to file <FileReference of the file/directory being opened> but the file has no object id in it”.
  + MSG SHLOG REPARSE\_INDEX\_ENTRY\_WITH\_NO\_REPARSE\_FRS
    - “The reparse point index entry in file <FileReference of the reparse point index table file> points to file <FileReference of the file/directory being opened> but the file has no reparse point in it”.
  + MSG SHLOG INCORRECT\_SEQUENCE\_NUMBER
    - “The [sequence number](http://msdn.microsoft.com/en-us/library/bb470124(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened> is incorrect”.
  + MSG SHLOG INCORRECT\_SEGMENT\_NUMBER
    - “The [segment number](http://msdn.microsoft.com/en-us/library/bb470211(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened> is incorrect”.
  + MSG SHLOG ATTR\_RECORD\_OFFSET\_TOO\_LARGE
    - “The [attribute record offset](http://msdn.microsoft.com/en-us/library/bb470124(VS.85).aspx) <the current value> is too large for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened>”.
  + MSG SHLOG ATTR\_RECORD\_LENGTH\_CANNOT\_BE\_ZERO
    - “The record length of [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened> should not be zero”.
  + MSG SHLOG ATTR\_RECORD\_LENGTH\_MISALIGNED
    - “The record length <the current value> of [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened> is not aligned”.
  + MSG SHLOG ATTR\_RECORD\_TOO\_LARGE
    - “The record length <the current value> is too large for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened>”.
  + MSG SHLOG INCORRECT\_FIRST\_FREE\_BYTE
    - “The first free byte, <the current value>, in file <FileReference of the file/directory being opened> is incorrect. The number of bytes free in the file record segment is 0 and the total length is <the current [BytesPerFileRecordSegment](http://msdn.microsoft.com/en-us/library/aa365256(VS.85).aspx) value of the specific NTFS volume>”.
  + MSG SHLOG ATTR\_RECORD\_LENGTH\_TOO\_SMALL
    - “The [attribute record length (RecordLength)](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> is too small for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value>”.
  + MSG SHLOG INVALID\_ATTR\_FORM\_CODE
    - “The [attribute form code (FormCode)](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> is invalid for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value>”.
  + MSG SHLOG ATTR\_SHOULD\_BE\_RESIDENT
    - “The [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> should be resident”.
  + MSG SHLOG INCORRECT\_STD\_INFO\_ATTR\_SIZE
    - “The standard information [attribute length (Form.Resident.ValueLength)](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> is incorrect”.
  + MSG SHLOG ATTR\_SHOULD\_NOT\_HAVE\_NAME
    - “Attribute [name (pointed by NameOffset)](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) is not allowed for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value>”.
    - Attributes of the following type should not have a name:
      * [$ATTRIBUTE\_LIST](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$OBJECT\_ID](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$SECURITY\_DESCRIPTOR](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$VOLUME\_NAME](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$VOLUME\_INFORMATION](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$REPARSE\_POINT](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$EA\_INFORMATION](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$EA](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx).
  + MSG SHLOG ATTR\_SHOULD\_NOT\_BE\_RESIDENT
    - “The [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> should not be resident”.
    - Attributes of the following type should not be resident:
      * [$INDEX\_ALLOCATION](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx).
  + MSG SHLOG INCORRECT\_ATTR\_NAME\_OFFSET
    - “The [attribute name offset (NameOffset)](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> is incorrect”.
  + MSG SHLOG NULL\_FOUND\_IN\_ATTR\_NAME
    - “The attribute [name (pointed by NameOffset)](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> contains unicode NULL”.
  + MSG SHLOG UNKNOWN\_ATTR\_TO\_ATTR\_DEF\_TABLE
    - “Unknown [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value>”.
    - Only the following are known attribute types:
      * [$STANDARD\_INFORMATION](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$ATTRIBUTE\_LIST](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$FILE\_NAME](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$OBJECT\_ID](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$SECURITY\_DESCRIPTOR](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$VOLUME\_NAME](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$VOLUME\_INFORMATION](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$DATA](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$INDEX\_ROOT](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$INDEX\_ALLOCATION](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$BITMAP](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$REPARSE\_POINT](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$EA\_INFORMATION](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$EA](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx);
      * [$LOGGED\_UTILITY\_STREAM](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx).
  + MSG SHLOG ATTR\_SHOULD\_NOT\_BE\_INDEXED
    - “The [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> should not be indexed”.
  + MSG SHLOG ATTR\_SHOULD\_BE\_INDEXED
    - “The [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> should be indexed”.
  + MSG SHLOG INDEXABLE\_ATTR\_SHOULD\_NOT\_HAVE\_NAME
    - “The indexable [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> should not have name”.
  + MSG SHLOG ATTR\_SHOULD\_BE\_NAMED
    - “The [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> should have a name”.
  + MSG SHLOG ATTR\_LENGTH\_TOO\_SMALL
    - “The [attribute length (Form.Resident.ValueLength)](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> is too small”.
  + MSG SHLOG ATTR\_LENGTH\_TOO\_BIG
    - “The [attribute length (Form.Resident.ValueLength)](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> is too big”.
  + MSG SHLOG INCORRECT\_RESIDENT\_ATTR
    - “The resident attribute for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> is incorrect. The attribute has value of length ([Form.Resident.ValueLength](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) <the current value> and offset ([Form.Resident.ValueOffset](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) <the current value>. The attribute length ([RecordLength](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) is <the current value>”.
  + MSG SHLOG RESIDENT\_ATTR\_COLLISION
    - “The resident attribute name is colliding with the resident value for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value>. The [attribute name offset (NameOffset)](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) is <the current value>, length ([NameLength](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) <the current value>, and the attribute value offset ([Form.Resident.ValueOffset](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) is <the current value>”.
  + MSG SHLOG NON\_RESIDENT\_ATTR\_HAS\_BAD\_MAPPING\_PAIRS\_OFFSET
    - “The mapping pairs offset ([MappingPairsOffset](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) <the current value> for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> exceeded the attribute length ([RecordLength](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) <the current value>”.
  + MSG SHLOG NON\_RESIDENT\_ATTR\_HAS\_UNALIGNED\_MAPPING\_PAIRS\_OFFSET
    - “The mapping pairs offset ([MappingPairsOffset](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) <the current value> for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> is not quad aligned”.
  + MSG SHLOG NON\_RESIDENT\_ATTR\_MAPPING\_PAIRS\_OFFSET\_TOO\_SMALL
    - “The mapping pairs offset ([MappingPairsOffset](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) <the current value> for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> is too small”.
  + MSG SHLOG NON\_RESIDENT\_ATTR\_COLLISION
    - “The attribute name is colliding with the mapping pairs for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value>. The attribute name offset [attribute name offset (NameOffset)](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) is <the current value>, length ([NameLength](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) <the current value>, and the mapping pairs offset ([MappingPairsOffset](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) is <the current value>”.
  + MSG SHLOG INVALID\_NON\_RESIDENT\_ATTR\_SIZES
    - “The non resident [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value>. The valid data length ([ValidDataLength](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) is <the current value>, file size ([FileSize](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) <the current value>, and allocated length (AllocatedLength) <the current value>”.
  + MSG SHLOG INVALID\_NON\_RESIDENT\_ATTR\_TOTAL\_ALLOC
    - “The total allocated size ([TotalAllocated](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) <the current value> for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> is larger than the allocated length ([AllocatedLength](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) <the current value>”.
  + MSG SHLOG INVALID\_NON\_RESIDENT\_ATTR\_TOTAL\_ALLOC\_BLOCK
    - “The allocated length ([AllocatedLength](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) <the current value> is not in multiple of <the current value of [CompressionUnit](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx)> for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value>”.
  + MSG SHLOG BAD\_FILE\_NAME\_LENGTH\_IN\_FILE\_NAME\_VALUE
    - “The file name length is zero for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value>”.
  + MSG SHLOG INCONSISTENCE\_FILE\_NAME\_VALUE
    - “The [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> is inconsistence. The attribute value length ([Form.Resident.ValueLength](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) is <the current value>”.
  + MSG SHLOG NULL\_FOUND\_IN\_FILE\_NAME\_OF\_FILE\_NAME\_VALUE
    - “The file name in file name value (pointed by [Form.Resident.ValueOffset](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx)) in [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> contains unicode NULL”.
  + MSG SHLOG ATTR\_LOWEST\_VCN\_IS\_NOT\_ZERO
    - “The [lowest virtual cluster number (VCN)](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx), <the current value>, is not zero for [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened>”.
  + MSG SHLOG INVALID\_FILE\_ATTR
    - “The [file attributes flag](http://msdn.microsoft.com/en-us/library/ms791554.aspx) <the current value> in file <FileReference of the file/directory being opened> is incorrect”.
  + MSG SHLOG MISSING\_STANDARD\_INFO
    - “The standard information attribute in file <FileReference of the file/directory being opened> is missing”.
  + MSG SHLOG MISSING\_FILE\_NAME\_INDEX\_PRESENT\_BIT
    - “The [file name index present bit (FILE\_FILE\_NAME\_INDEX\_PRESENT)](http://msdn.microsoft.com/en-us/library/bb470124(VS.85).aspx) is not set for file <FileReference of the file/directory being opened>”.
  + MSG SHLOG FILE\_NAME\_INDEX\_PRESENT\_BIT\_SET
    - “The [file name index present bit (FILE\_FILE\_NAME\_INDEX\_PRESENT)](http://msdn.microsoft.com/en-us/library/bb470124(VS.85).aspx) is set in file <FileReference of the file/directory being opened> but there is no file name index”.
  + MSG SHLOG CANNOT\_FIND\_UNNAMED\_DATA\_ATTR
    - “The unnamed data attribute in file <FileReference of the file/directory being opened> is missing”.
  + MSG SHLOG MISSING\_SPARSE\_FLAG\_IN\_STD\_INFO
    - “The [sparse flag (FILE\_ATTRIBUTE\_SPARSE\_FILE)](http://msdn.microsoft.com/en-us/library/ms791526.aspx) in the standard information attribute in file <FileReference of the file/directory being opened> is not set”.
  + MSG SHLOG NTFS\_SPARSE\_FLAG\_SET\_IN\_STD\_INFO
    - “The [sparse flag (FILE\_ATTRIBUTE\_SPARSE\_FILE)](http://msdn.microsoft.com/en-us/library/ms791526.aspx) in the standard information attribute in file <FileReference of the file/directory being opened> should not be set”.
  + MSG SHLOG MISSING\_ENCRYPTED\_FLAG\_IN\_STD\_INFO
    - “The [encrypted flag (FILE\_ATTRIBUTE\_ENCRYPTED)](http://msdn.microsoft.com/en-us/library/ms791526.aspx) in standard information attribute in file <FileReference of the file/directory being opened> is not set”.
  + MSG SHLOG MISSING\_REPARSE\_POINT\_FLAG\_IN\_STD\_INFO
    - “The [reparse flag (FILE\_ATTRIBUTE\_REPARSE\_POINT)](http://msdn.microsoft.com/en-us/library/ms791526.aspx) in standard information attribute in file <FileReference of the file/directory being opened> is not set”.
  + MSG SHLOG REPARSE\_POINT\_FLAG\_SET\_IN\_STD\_INFO
    - “The [reparse flag (FILE\_ATTRIBUTE\_REPARSE\_POINT)](http://msdn.microsoft.com/en-us/library/ms791526.aspx) in standard information attribute in file <FileReference of the file/directory being opened> should not be set”.
  + MSG SHLOG BAD\_MAPPING\_PAIRS
    - “The attribute [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened> has bad [mapping pairs at attribute offset (MappingPairsOffset)](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value>”.
* The following are reports of specific corruption fixing actions having been taken (appearing between the start report and the end report of a specific self healing repair run).
  + MSG SHLOG DELETING\_GENERIC\_INDEX\_ENTRY
    - “Deleting an index entry from index <FileReference[[37]](#footnote-38) of the file/directory being opened> of file <ParentFileReference of the parent directory>”.
  + MSG SHLOG BAD\_FRS
    - “Deleting corrupt file record segment <FileReference of the file/directory being opened>”.
  + MSG SHLOG DELETING\_INDEX\_ENTRY
    - “Deleting index entry <the file name of the [file name attribute](http://msdn.microsoft.com/en-us/library/bb470123(VS.85).aspx)> in index <ParentFileReference of the parent directory> of file <FileReference of the file/directory being opened>”.
  + MSG SHLOG CLEARING\_IN\_USE
    - “Clearing the in use bit for file record <FileReference of the file/directory being opened>”.
  + MSG SHLOG REPAIRING\_SEQUENCE\_NUMBER
    - “Repairing the [sequence number](http://msdn.microsoft.com/en-us/library/bb470124(VS.85).aspx) for file record <FileReference of the file/directory being opened>”.
  + MSG SHLOG REPAIRING\_SEGMENT\_NUMBER
    - “Repairing the [segment number](http://msdn.microsoft.com/en-us/library/bb470211(VS.85).aspx) for file record <FileReference of the file/directory being opened>”.
  + MSG SHLOG REPAIRING\_FIRST\_FREE\_BYTE
    - “Repairing the first free byte for file record <FileReference of the file/directory being opened>”.
  + MSG SHLOG DELETING\_ATTRIBUTE
    - “Deleting [attribute of type](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> and [instance tag](http://msdn.microsoft.com/en-us/library/bb470039(VS.85).aspx) <the current value> in file <FileReference of the file/directory being opened>”.
  + MSG SHLOG REPAIRING\_FILE\_ATTRIBUTES
    - “Repairing the file attributes for file record <FileReference of the file/directory being opened>”.
  + MSG SHLOG REPAIRING\_FILE\_RECORD\_FLAGS
    - Repairing the flags for file record <FileReference of the file/directory being opened>.
  + MSG SHLOG ADD\_ATTRIBUTE
    - Adding attribute of type code “[$DATA](http://msdn.microsoft.com/en-us/library/bb470038(VS.85).aspx)” for file record <FileReference of the file/directory being opened>.

In the case where a corruption repair action has been taken, but there is a possibility of data lost, the Windows OS NTFS sends a dialog box to the interactive user, warning about the following device IO error that indicates the specific file where a data lost may have occurred.

* [STATUS\_DATA\_LOST\_REPAIR](http://msdn.microsoft.com/en-us/library/cc704588.aspx) “Windows discovered a corruption in the file <the name of the file where the corruption was detected and repaired>. This file has now been repaired. Please check if any data in the file was lost because of the corruption”.

The Windows OS NTFS supports the generation of a maximum of 10 Event ID 130 or Event ID 131 audit records per minute. If the rate of the “Event ID 130 or Event ID 131 audit record” generation reaches 10 per minute, the Windows OS NTFS suspends the generation of the Event ID 130 or Event ID 131 audit record until the rate falls below 10 per minute again. Instead, the Windows OS NTFS generates the following Event ID 132 soft audit record in the soft audit storage[[38]](#footnote-39), indicating the specific NTFS partition/volume where the repair events are occurring.

* Event ID 132 (NTFS\_ETW\_REPAIR\_EVENT\_SUSPENDED) “Too many repair events has occurred in a short period of time. Temporarily suspending posting of further repair events”.

After the “Event ID 130 or Event ID 131 audit record” generation rate falls below 10 per minute, the Windows OS NTFS generates the following Event ID 133 soft audit record in the soft audit storage, indicating the specific NTFS partition/volume where the repair events are occurring and the number of skipped audit records of Event ID 130 or Event ID 131.

* Event ID 133 (NTFS\_ETW\_REPAIR\_EVENT\_RESUMED) “Skipped posting of <N number> repair events. Repair event posting will now be resumed”.

The Windows OS NTFS currently supports the following self healing policies, which may be configured by an authorized administrator using the [FSCTL\_SET\_REPAIR Control Code](http://msdn.microsoft.com/en-us/library/aa964904(VS.85).aspx) against a specific NTFS partition/volume:

* [SET\_REPAIR\_ENABLED](http://msdn.microsoft.com/en-us/library/aa964904(VS.85).aspx) (which is the default)
  + Self-healing is turned on;
* [SET\_REPAIR\_DISABLED\_AND\_BUGCHECK\_ON\_CORRUPT](http://msdn.microsoft.com/en-us/library/aa964904(VS.85).aspx)
  + If set, and the “NtfsBugCheckOnCorrupt” registry key value under the following registry key is also set to 1 (one), the Windows OS NTFS issues a system stop error NTFS\_FILE\_SYSTEM “0x24” (also called a bugcheck or blue screen) if the specific NTFS partition/volume is corrupted.
    - “HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\FileSystem”.

The Windows OS NTFS also clears [SET\_REPAIR\_DISABLED\_AND\_BUGCHECK\_ON\_CORRUPT](http://msdn.microsoft.com/en-us/library/aa964904(VS.85).aspx) setting before it issues the system stop error to avoid repeated reboot cycles. This setting is useful for capturing a dump file of the system so that the problem can better be analyzed by authorized support personnel.

Regardless of the above setting, the Windows OS NTFS generates the following Event ID 55 soft audit record, indicating the corrupted file where possible, in the soft audit storage, when a specific NTFS partition/volume encounters a corruption where the [self healing NTFS](http://download.microsoft.com/download/8/1/E/81EFEE63-F192-4BD7-AE85-2DDBDA888EEE/WS2008RAS.docx) is unable to repair automatically.

* [Event ID 55](http://www.microsoft.com/technet/support/ee/transform.aspx?ProdName=Windows+Operating+System&ProdVer=5.2&EvtID=55&EvtSrc=ntfs&LCID=1033) (IO\_FILE\_SYSTEM\_CORRUPT\_WITH\_NAME) “The file system structure on the disk is corrupt and unusable. Please run the chkdsk utility on the volume <Volume ID>”.

The Windows OS NTFS also sends a dialog box to the interactive user, warning about one of the following device IO errors that indicate why the user’s IO request has failed.

* [STATUS\_DISK\_CORRUPT\_ERROR](http://msdn.microsoft.com/en-us/library/cc704588.aspx) “The file system structure on the disk is corrupt and unusable. Please run the Chkdsk utility on the volume <Volume ID>”.
* [STATUS\_FILE\_CORRUPT\_ERROR](http://msdn.microsoft.com/en-us/library/cc704588.aspx) “The file or directory <the name of the file/directory where the corruption is detected> is corrupt and unreadable. Please run the Chkdsk utility”.

## Addressing 5.4.1.2 “The OS shall use an integrity function provided by the encryption support system for the integrity protection of encrypted and stored data”

The Windows OS addresses the “5.4.1.2” requirement because of the availability of the Windows OS BitLocker™ full volume encryption capability for hard disk volumes, as described in the “Full volume encryption” section of this paper. As further explained in the Microsoft publication: <http://download.microsoft.com/download/0/2/3/0238acaf-d3bf-4a6d-b3d6-0a0be4bbb36e/BitLockerCipher200608.pdf>, the Windows OS BitLocker™ full volume encryption capability provides the “Poor-man's authentication” solution of “encrypt the data and trust to the fact that changes in the cipher-text do not translate to semantically sensible changes to the plaintext”, using the “AES-CBC + diffuser” encryption scheme on a per-(disk)sector basis. As stated in this Microsoft publication, a goal of the Windows OS BitLocker™ full volume encryption is that “an attacker cannot control or predict any aspect of the plaintext changes if he modifies or replaces the cipher-text of a sector”. We also note that the Microsoft publication especially explains why the use of a Message Authentication Code (MAC) for each block of data on the disk is not universally suitable in the expected Windows deployment scenarios.

## Addressing 5.4.1.3 “To the data of the OS stored on disk, the OS shall realize an “automatically check the integrity of files” function”

The “5.4.1.3” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “5.3.1.3” requirement “The OS shall verify during initial startup the integrity of executable code that implements access control and cryptographic functionality through the use of the OS system provided cryptographic services”.

For the executable data of the OS, the Windows OS similarly addresses the “5.4.1.3” requirement with an exception. Specifically, the Windows OS addresses the “5.4.1.3” requirement for specific Windows OS executable image files. As explained in the “Integrity check conducted by the Windows OS boot winload application” section of this paper, the following Windows OS executable image files are checked for their integrity before they are loaded into the memory.

* The Windows OS kernel (ntoskrnl.exe) as listed in the hardcoded “Microsoft Boot Images” list;
* The Windows OS code integrity library module (ci.dll) as listed in the hardcoded “Microsoft Boot Images” list;
* The BitLocker™ drive encryption filter driver (fvevol.sys) as listed in the hardcoded “Microsoft Boot Images” list;
* The Boot Video Driver (bootvid.dll) as listed in the hardcoded “Microsoft Boot Images” list;
* The Common Log Driver (clfs.sys) as listed in the hardcoded “Microsoft Boot Images” list;
* The Hardware Abstraction Layer Driver (hal.dll) as listed in the hardcoded “Microsoft Boot Images” list;
* The Kernel Debugger Serial Communication library (kdcom.dll) as listed in the hardcoded “Microsoft Boot Images” list;
* The Platform Specific Hardware Error Driver library (pshed.dll) as listed in the hardcoded “Microsoft Boot Images” list;
* The Software Licensing Support Driver library (spldr.sys) as listed in the hardcoded “Microsoft Boot Images” list;
* The Trusted Platform Module (TPM) Device Driver (tpm.sys) as listed in the hardcoded “Microsoft Boot Images” list;
* The Microcode Update Library (mcupdate.dll) as listed in the hardcoded “Microsoft Boot Images” list;
* An executable Windows OS image file, which is expected to run in the protected environment as a protected process;
* A Windows OS image file which need to be loaded in the address space of a protected process running in the protected environment;
* A Windows OS image file which is a kernel mode driver;
* A Windows OS image file which is a kernel mode driver in a hot-patching scenario;
* A Windows OS image file where the IMAGE\_ DLLCHARACTERISTICS\_ FORCE\_ INTEGRITY flag is set in the DllCharacteristics field of the file’s IMAGE\_OPTIONAL\_HEADER.

The other Windows OS executable image files (which are not listed in the above) are not checked for their integrity before they are loaded into the memory. They therefore represent a minor exception.

Please see the justification text for addressing the Commercial Grade OS Requirement Set “5.3.1.3” requirement for more detail.

For the non-executable data of the OS, the Windows OS similarly addresses the “5.4.1.3” requirement because of the “5.4.1.2” requirement which states that “The OS shall use an integrity function provided by the encryption support system for the integrity protection of encrypted and stored data”. Please see the justification text for addressing the “5.4.1.2” requirement in Appendix B for more detail.

## Addressing 5.4.1.4 “To the data of the OS stored on disk, the OS shall realize an “automatically check the integrity of disk surface” function”

The Windows OS addresses the “5.4.1.4” requirement as follows.

When the “/r -- Locates bad sectors and recovers free sectors” command option is specified, the [chkdsk tool](http://technet.microsoft.com/en-us/library/cc730714.aspx) sends the [IOCTL\_DISK\_VERIFY](http://msdn.microsoft.com/en-us/library/aa365193(VS.85).aspx) to the specific NTFS partition/volume in an attempt to identify the free usable sectors. When the Windows OS Disk Driver (disk.sys) receives this [IOCTL\_DISK\_VERIFY](http://msdn.microsoft.com/en-us/library/aa365193(VS.85).aspx), it issues a [SCSI “Verify” command](http://ldkelley.com/SCSI2/SCSI2/SCSI2-09.html#9.2.19) to the target physical device for the specific NTFS partition/volume”.

For the written sectors, the Windows OS relies on the default capability of the [self healing NTFS](http://download.microsoft.com/download/8/1/E/81EFEE63-F192-4BD7-AE85-2DDBDA888EEE/WS2008RAS.docx), as explained in the “Self healing Windows OS NTFS” section under Appendix B of this paper, for the automatic checking of the integrity associated with the data of the written sectors.

## Addressing 5.4.1.5 “To the data of the OS stored on disk, the OS shall realize a “check and diagnose as well as recover from errors on the disk at any time” function”

The Windows OS addresses the “5.4.1.5” requirement because of the default capability of the [self healing NTFS](http://download.microsoft.com/download/8/1/E/81EFEE63-F192-4BD7-AE85-2DDBDA888EEE/WS2008RAS.docx), as explained in the “Self healing Windows OS NTFS” section under Appendix B of this paper.

In addition, the “[chkdsk command line tool](http://technet.microsoft.com/en-us/library/cc730714.aspx)” supports the following command options, other than just checking the file system and file system metadata of a target NTFS partition/volume for logical and physical errors.

* “/f -- Fixes errors on the disk, which the disk must be locked”.
* “/r -- Locates bad sectors and recovers readable information”.
* “/b -- Clears the list of bad clusters on the volume and rescans all allocated and free clusters for errors”.

## Addressing 5.4.1.6 “To the data of the OS stored on disk, the OS shall realize a “repair sector intertwining and sectors lost” function”

The Windows OS addresses the “5.4.1.6” requirement because of the default capability of the [self healing NTFS](http://download.microsoft.com/download/8/1/E/81EFEE63-F192-4BD7-AE85-2DDBDA888EEE/WS2008RAS.docx), as explained in the “Self healing Windows OS NTFS” section under Appendix B of this paper.

In addition, the “[chkdsk command line tool](http://technet.microsoft.com/en-us/library/cc730714.aspx)” supports the following command options, other than just checking the file system and file system metadata of a target NTFS partition/volume for logical and physical errors.

* “/f -- Fixes errors on the disk, which the disk must be locked”.
* “/r -- Locates bad sectors and recovers readable information”.
* “/b -- Clears the list of bad clusters on the volume and rescans all allocated and free clusters for errors”.

We note that the Windows OS also supports [redundant array of independent disks (RAID) for achieving high availability](http://technet.microsoft.com/en-us/library/cc947587.aspx). Specifically, the following RAID configurations are provided through the Disk Management snap-in (dmdskmgr.dll)”.

* Striped volumes use RAID-0, which stripes data across multiple disks. RAID-0 does not offer fault tolerance, but it does offer increased performance.
* Mirrored volumes use RAID-1, which provides redundancy by creating two identical copies of a volume.
* RAID-5 volumes use RAID-5, which stripes parity information across multiple disks. This parity information can be used to recreate data stored on a failed disk.

## Addressing 5.4.1.7 “To the data of the OS stored on disk, the OS shall realize a “move data to good sectors” function”

The Windows OS addresses the “5.4.1.7” requirement because the Windows OS [Disk Defragmenter](http://support.microsoft.com/kb/942092) (dfrgui.exe) move data to good sectors as it rearranges the data on the target NTFS partition/volume and reunites fragmented files.

Furthermore, the Windows OS [Disk Defragmenter](http://support.microsoft.com/kb/942092) supports scheduling of its activities.

## Addressing 5.4.1.8 “To the data of the OS stored on disk, the OS shall realize a “hard disk data backup and restoration” function”

The Windows OS addresses the “5.4.1.8” requirement because the Windows OS backup and restore is based on the Windows OS Volume Shadow Copy Service (VSSVC.exe) and the Windows OS Block Level Backup Engine Service (wbengine.exe). As explained in the following “Windows OS Backup and Restore” section, the Windows OS services and applications do not need to be interrupted when their data need to be copied for backup or restore”.

### Windows OS Backup and Restore

The Windows OS [Block Level Backup Engine Service](http://technet.microsoft.com/en-us/library/cc766422.aspx) (wbengine.exe) performs block level backup and recovery of data. A number of Windows OS backup and restore tools use this service. These tools are described as follows.

* Windows Vista [Backup and Restore Center](http://www.microsoft.com/windows/windows-vista/features/backup.aspx) (BRCpl.dll)
  + It includes the following capabilities:
    - [Automatic scheduled and network file backup](http://www.microsoft.com/windows/windows-vista/features/backup.aspx)
      * It provides the ability to schedule automatic file backup to a remote, local, or external target location;
    - [Windows Complete PC Backup and Restore](http://www.microsoft.com/windows/windows-vista/features/backup.aspx)
      * It provides an image-based backup of the Windows PC environment, including the operating system, installed programs, system and user settings, and data files so that the same Windows PC environment may be restored in the [Windows Recovery Environment (WinRE)](http://technet.microsoft.com/en-us/library/cc749147.aspx) via the “Windows Complete PC Restore” tool (bmrui.exe).
* Windows Server 2008 Backup Microsoft Management Console (MMC) snap-in (blbmmc.dll)
  + It includes the following application wizards:
    - [Backup Once Wizard](http://technet.microsoft.com/en-us/library/cc770593.aspx#BKMK_Windows_Server_Backup_overview)
      * It creates a one-time supplemental backup of all or selected volumes (including files and applications) and the current system state;
    - [Scheduled Backup Wizard](http://technet.microsoft.com/en-us/library/cc770593.aspx#BKMK_Windows_Server_Backup_overview)
      * It provides the ability to schedule automatic backup all or selected volumes (including files and applications) and the system state to a remote, local, or external target location;
    - [Recovery Wizard](http://technet.microsoft.com/en-us/library/cc770593.aspx#BKMK_Windows_Server_Backup_overview)
      * It recovers files, folders, applications, and volumes;
    - [Catalog Recovery Wizard](http://technet.microsoft.com/en-us/library/cc770593.aspx#BKMK_Windows_Server_Backup_overview)
      * It recovers a backup catalog, which contains details about a specific backup.
* The “[Wbadmin.exe](http://technet.microsoft.com/en-us/library/cc754015.aspx)” command tool with the following subcommands.
  + [Wbadmin enable backup](http://technet.microsoft.com/en-us/library/cc742130.aspx)
    - It configures or modifies a daily backup run schedule.
  + [Wbadmin disable backup](http://technet.microsoft.com/en-us/library/cc742068.aspx)
    - It disables a scheduled daily backup run.
  + [Wbadmin start backup](http://technet.microsoft.com/en-us/library/cc742083.aspx)
    - It runs a one-time backup run with specific new settings or with the settings from the daily backup schedule.
  + [Wbadmin stop job](http://technet.microsoft.com/en-us/library/cc742064.aspx)
    - It cancels the backup or recovery operation that is currently running.
  + [Wbadmin get items](http://technet.microsoft.com/en-us/library/cc742041.aspx)
    - It lists the items included in a specific backup run.
  + [Wbadmin start recovery](http://technet.microsoft.com/en-us/library/cc742070.aspx)
    - It runs a recovery for a specific volume, application, or file based on the specified settings.
  + [Wbadmin get versions](http://technet.microsoft.com/en-us/library/cc742116.aspx)
    - It lists details about the available backups stored in the specified location.
  + [Wbadmin get status](http://technet.microsoft.com/en-us/library/cc742050.aspx)
    - It reports the status of the current backup or recovery run.
  + [Wbadmin start systemstaterecovery](http://technet.microsoft.com/en-us/library/cc742035.aspx)
    - It performs a system state recovery to a target Windows OS machine from a specific backup location.
  + [Wbadmin start systemstatebackup](http://technet.microsoft.com/en-us/library/cc742124.aspx)
    - It creates a system state backup of the local computer and stores the backup at the specified location.
  + [Wbadmin delete systemstatebackup](http://technet.microsoft.com/en-us/library/cc742081.aspx)
    - It deletes the specified system state backups.
  + [Wbadmin start sysrecovery](http://technet.microsoft.com/en-us/library/cc742118.aspx)
    - It performs in the [Windows Recovery Environment (WinRE)](http://technet.microsoft.com/en-us/library/cc749147.aspx) a recovery run of the full system (at least all the volumes that contain the operating system's state) from a specific backup location.
  + [Wbadmin restore catalog](http://technet.microsoft.com/en-us/library/cc742142.aspx)
    - It recovers a backup catalog from the specified storage location in the case where the backup catalog residing in the local computer has been corrupted.
  + [Wbadmin delete catalog](http://technet.microsoft.com/en-us/library/cc742154.aspx)
    - It deletes the backup catalog residing in the local computer.

The Windows OS [Block Level Backup Engine Service](http://technet.microsoft.com/en-us/library/cc766422.aspx) (wbengine.exe) is implemented as a DCOM server running in the security context of local service. It handles requests from a subject using one of the above Windows OS backup and restore tools through its DCOM interfaces. The Windows OS [Block Level Backup Engine Service](http://technet.microsoft.com/en-us/library/cc766422.aspx) (wbengine.exe) enforces the necessary access security in its DCOM interfaces. Only a member of the “Backup Operators” or the “Administrators” group is allowed to gain access to the DCOM interfaces. The Windows OS [Block Level Backup Engine Service](http://technet.microsoft.com/en-us/library/cc766422.aspx) (wbengine.exe) also enforces the administrator configured Windows OS Backup and Restore Policies.

#### Windows OS Backup and Restore Policies

The following Windows OS Backup and Restore Policies may be configured by an administrator.

* The Windows Vista Backup and Restore Policies
  + “Prevent the user from running the Backup Status and Configuration program”
    - It lets an administrator disable the Backup Status and Configuration program, which links to the file backup, file restore, and Complete PC Backup applications and shows backup status.
    - If this policy is enabled, a user cannot start the Backup Status and Configuration program.
    - By default, this policy is not enabled.
  + “Prevent backing up to local disks”
    - This policy lets an administrator prevent users from selecting a local disk (internal or external) for storing file backups.
    - If this policy is enabled, a user is blocked from selecting a local disk as a file backup location.
    - By default, this policy is not enabled.
  + “Prevent backing up to network shared folder”
    - This policy lets an administrator prevent users from selecting a network shared folder for storing file backups.
    - If this policy is enabled, a user is blocked from selecting a network shared folder as a file backup location.
    - By default, this policy is not enabled.
  + “Prevent backing up to optical media (CD/DVD)”
    - This policy lets an administrator prevent users from selecting optical media (CD/DVD) for storing file backups.
    - If this policy is enabled, a user is blocked from selecting optical media as a file backup location.
    - By default, this policy is not enabled.
  + “Turn off backup configuration”
    - This policy lets an administrator disable file backup functionality.
    - If this policy is enabled, the file backup program is disabled.
    - By default, this policy is not enabled.
  + “Turn off restore functionality”
    - This policy lets an administrator disable file restore functionality.
    - If this policy is enabled, the file restore program is disabled.
    - By default, this policy is not enabled.
  + “Turn off Complete PC Backup functionality”
    - This policy lets an administrator disable Complete PC Backup functionality.
    - If this policy is enabled, the Complete PC Backup program is disabled.
    - By default, this policy is not enabled.
* The Windows Server 2008 Backup and Restore Policies
  + “Allow only system backup”
    - This policy lets an administrator manage whether backups of only system volumes are allowed or both OS and data volumes can be backed up.
    - If this policy is enabled, a member of the “Backup Operators” or the “Administrators” group can backup only volumes hosting OS components and no data only volumes can be backed up. Otherwise, backups can include both system or data volumes.
    - By default, this policy is not enabled.
  + “Disallow locally attached storage as backup target”
    - This policy lets an administrator manage whether backups of a machine can run to locally attached storage or not.
    - If this policy is enabled, a member of the “Backup Operators” or the “Administrators” group cannot use Windows Server Backup to run backups to a locally attached storage or disk. Otherwise, there is no restriction on locally attached storage or disk being backup target.
    - By default, this policy is not enabled.
  + “Disallow network as backup target”
    - This policy lets an administrator manage whether backups of a machine can run to a network share or not.
    - If this policy is enabled, a member of the “Backup Operators” or the “Administrators” group cannot use Windows Server Backup to run backups to a network share. Otherwise, there is no restriction on network share being backup target.
    - By default, this policy is not enabled.
  + “Disallow optical media as backup target”
    - This policy lets an administrator manage whether backups of a machine can run to an optical media or not.
    - If this policy is enabled, a member of the “Backup Operators” or the “Administrators” group cannot use Windows Server Backup to run backups to an optical media. Otherwise, there is no restriction on optical media being backup target.
    - By default, this policy is not enabled.
  + “Disallow run-once backups”
    - This policy lets an administrator manage whether run-once backups of a machine can be run or not.
    - If this policy is enabled, a member of the “Backup Operators” or the “Administrators” group cannot use Windows Server Backup to run non-scheduled run-once backups. Otherwise, there is no restriction on running run-once backups.
    - By default, this policy is not enabled.

#### Windows OS Volume Shadow Copy Service underlying Windows OS Block Level Backup Engine Service

The Windows OS [Volume Shadow Copy Service (VSS)](http://msdn.microsoft.com/en-us/library/aa384624(VS.85).aspx) [VSSVC.exe] provides the underlying volume shadowing capability for the Windows OS [Block Level Backup Engine Service](http://technet.microsoft.com/en-us/library/cc766422.aspx) (wbengine.exe), as a VSS application needs to conduct its backup and restore services. The Windows OS VSS is responsible for the following roles.

* Coordination of the activities of VSS [providers](http://msdn.microsoft.com/en-us/library/aa384594(VS.85).aspx), [writers](http://msdn.microsoft.com/en-us/library/aa384993(VS.85).aspx), and [requesters](http://msdn.microsoft.com/en-us/library/aa384596(VS.85).aspx) in the creation and use of shadow copies.
* Furnishing the [default system provider](http://msdn.microsoft.com/en-us/library/aa384594(VS.85).aspx).
* Implementing low-level driver functionality (through the “volsnap.sys” kernel mode driver) necessary for any provider to work.

A shadow copy of a volume duplicates all the data held on that volume at one well-defined instant in time. The Windows OS VSS identifies each shadow copy by a persistent GUID. Once a shadow copy is created, there are effectively two images of the shadow-copied volume available to the system:

* the original volume, which can be accessed conventionally;
* the copied data, which can be accessed through the VSS API.

This allows two sets of activities to take place at the same time:

* Ordinary applications on the system can quickly continue or resume using the original volume, updating data on the disk;
* Applications (such as the Windows OS [Block Level Backup Engine Service](http://technet.microsoft.com/en-us/library/cc766422.aspx)) that are using the VSS requester API to access the shadow-copied volume can perform backups or similar operations.

[VSS writers](http://msdn.microsoft.com/en-us/library/aa384993(VS.85).aspx) are Windows OS applications or services that store persistent information on disk and that cooperate with providers and requesters through the shadow copy interface. During backup operations, VSS writers ensure that their data is quiescent and stable (so that it is suitable for shadow copy and backup). VSS writers collaborate with restores by unlocking files when possible and indicating alternate locations when necessary. As a result, a VSS writer does not need to be interrupted when its data need to be copied for backup or restore. The VSS writer can continue to serve its clients while the backup of its data is occurring. The following Windows OS services are [VSS writers](http://msdn.microsoft.com/en-us/library/aa384993(VS.85).aspx) starting with the [CVssWriter::Initialize()](http://msdn.microsoft.com/en-us/library/aa381543(VS.85).aspx) or [CVssWriterEx::InitializeEx()](http://msdn.microsoft.com/en-us/library/aa381528(VS.85).aspx) function.

* Virtual Machine Management (vmms.exe);
* WMI Core Service (WMIsvc.dll);
* Microsoft Failover Cluster Service (clussvc.dll);
* Microsoft DFS Replication Service (dfsrS.exe) on Windows Server 2008;
* Microsoft DFS Replication Service (dfsr.exe) on Windows Vista;
* File Server Resource Manager (FSRM) service (srmsvc.dll);
* COM+ Runtime service (catsrv.dll);
* Removable Storage Service (ntmssvc.dll);
* File Replication Service (ntfrs.exe);
* Windows IIS Process Activation Service Process Model (apphostsvc.dll);
* IIS meta base service (iisadmin.dll);
* Internet Authentication Service Data Store (iasdatastore.dll);
* Windows OS registry;
* Cryptographic Services database (cryptsvc.dll);
* Windows OS Active Directory;
* Windows OS Certificate Server (certsrv.exe);
* Background Intelligent Transfer Service (qmgr.dll);
* Windows Deployment Services (WDS) Server (wdssrv.dll);
* DHCP Server Service (dhcpssvc.dll);
* Kernel Transaction Manager for Distributed Transaction Coordinator (msdtctm.dll).

These Windows OS services do not need to be interrupted when their data need to be copied for backup or restore as these OS services have the ability to implement the following methods of CVssWriter or CVssWriterEx. These methods are:

* [CVssWriter::OnIdentify()](http://msdn.microsoft.com/en-us/library/aa381564(VS.85).aspx);
* [CVssWriterEx::OnIdentifyEx()](http://msdn.microsoft.com/en-us/library/aa381529(VS.85).aspx);
* [CVssWriter::OnPrepareBackup()](http://msdn.microsoft.com/en-us/library/aa381571(VS.85).aspx);
* [CVssWriter::OnBackupComplete()](http://msdn.microsoft.com/en-us/library/aa381557(VS.85).aspx);
* [CVssWriter::OnBackupShutdown()](http://msdn.microsoft.com/en-us/library/aa381559(VS.85).aspx);
* [CVssWriter::OnPrepareSnapshot()](http://msdn.microsoft.com/en-us/library/aa381574(VS.85).aspx);
* [CVssWriter::OnPostSnapshot()](http://msdn.microsoft.com/en-us/library/aa381568(VS.85).aspx);
* [CVssWriter::OnAbort()](http://msdn.microsoft.com/en-us/library/aa381552(VS.85).aspx);
* [CVssWriter::OnFreeze()](http://msdn.microsoft.com/en-us/library/aa381563(VS.85).aspx);
* [CVssWriter::OnThaw()](http://msdn.microsoft.com/en-us/library/aa381579(VS.85).aspx);
* [CVssWriter::OnPreRestore()](http://msdn.microsoft.com/en-us/library/aa381577(VS.85).aspx);
* [CVssWriter::OnPostRestore()](http://msdn.microsoft.com/en-us/library/aa381566(VS.85).aspx).

## Addressing 5.4.1.9 “To the data of the OS stored on disk, the OS shall realize a “data compression on hard disk” function”

The Windows OS addresses the “5.4.1.9” requirement because the [Windows OS NTFS supports a lossless compression](http://msdn.microsoft.com/en-us/library/aa364219(VS.85).aspx) by implementing the Lempel-Ziv compression.

We note that the Windows OS EFS protection and the Windows OS NTFS compression are mutual exclusive in the Windows OS NTFS.

## Addressing 5.4.1.10 “To the data of the OS stored on disk, the OS shall realize a “backup on hard disk and restore if necessary” function”

The Windows OS addresses the “5.4.1.10” requirement because the Windows OS backup and restore is based on the Windows OS Volume Shadow Copy Service (VSSVC.exe) and the Windows OS Block Level Backup Engine Service (wbengine.exe). As explained in the “Windows OS Backup and Restore” section under Appendix B of this paper, the Windows OS services and applications do not need to be interrupted when their data need to be copied for backup or restore”.

## Addressing 5.4.1.11 “For data to be stored on disk, the OS data-writing operation shall be designed with an “invert back” requirement, where the data-writing operation can be inverted back in case of abnormalities”

The Windows OS addresses the “5.4.1.11” requirement because of the availability of [Transactional NTFS](http://msdn.microsoft.com/en-us/library/bb968806(VS.85).aspx) in the Windows OS. Transactional NTFS (TxF) allows file operations on an NTFS file system volume to be performed in a transaction. It supports application developers and administrators to gracefully handle errors and preserve data integrity. TxF also participates in distributed transactions that the [Distributed Transaction Coordinator (DTC)](http://msdn.microsoft.com/en-us/library/ms684146(VS.85).aspx) coordinates in the following manners:

* Transactions that span multiple data stores, for example, a single transaction for file and SQL operations;
* Transactions that span multiple computers, for example, a single transaction for file updates on multiple computers.

The Windows OS supports TxF in the following scenarios.

* Updating a File
  + The file is opened in transacted mode. The necessary updates are made following by a commitment of the transaction. If the system fails during the file update, then TxF automatically restores the file to the state that it had before the file update began, which avoids file corruption.
* Multi-File Updates
  + Multiple files are opened in a single transaction. If the system fails during the update of the files, then TxF automatically restores the files to the state that they had before the file update began, which avoids file corruption and inconsistencies among the files.
* Isolates concurrent transactions
  + If a transactional reader subject opens a file for a transactional read while the single transactional writer subject has the same file open for a transactional update, TxF isolates the effects of the two transactions from one another. The transactional reader subject always views a single, consistent version of the file, even while that file is in the process of being updated by the transactional writer subject.
* Creating a new file or directory
  + A file or directory that is created in a transaction is not visible to anything outside the current transaction.
* Deleting a new file or directory
  + A file or directory that is deleted by calling DeleteFileTransacted() is visible to all outside readers. All transacted handles to the file or directory must be closed before the end of the transaction. If the handles are not closed, the delete does not occur. If a directory is deleted, files within that directory are transactionally locked.
* Renaming a new file or directory
  + A file can be renamed (i.e. moved) as a transactional operation.
* Reparse points
  + Changes to reparse points are transacted. Assigning a new reparse point to a file or changing/removing an existing reparse point of a file in a transaction is not visible to the other transactions.
* Named streams
  + Named streams are transactional, but locking is done at the file level, not the stream level.
* Memory mapped files
  + Transactions described in the above work on memory mapped files, but a user need to use [FlushViewOfFile()](http://msdn.microsoft.com/en-us/library/aa366563(VS.85).aspx) before committing a transaction on a memory-mapped file.
* Encrypted file systems (EFS) protected files
  + Transactions described in the above work on EFS protected files on Windows Vista SP1 and Windows Server 2008.
  + However, encrypting or decrypting a file is not transacted.
* File compression
  + Transactions described in the above work on compressed files.
  + However, compressing or decompressing a file is not transacted.

On the other hand, the following scenarios are not supported in the Windows OS.

* Transactions on network volumes, for example on file shares. TxF is not supported by the [CIFS/SMB](http://msdn.microsoft.com/en-us/library/aa302188.aspx) protocols.
* Transacted operations against files cached by [client side caching for offline files](http://msdn.microsoft.com/en-us/library/cc296090(VS.85).aspx).
* File access using [file IDs](http://msdn.microsoft.com/en-us/library/aa365432(VS.85).aspx).
* Files needing to be opened for an extended period of time (days or weeks).
* Any file system other than NTFS.
* EFS encryption of TxF CLFS logs, the Txf Old Page Stream (TOPS) file, or the “$Txf” directory.
* Reading a raw encrypted file using the [ReadEncryptedFileRaw()](http://msdn.microsoft.com/en-us/library/aa365466(VS.85).aspx) within a transaction.
* Writing a raw encrypted file using the [WriteEncryptedFileRaw()](http://msdn.microsoft.com/en-us/library/aa365466(VS.85).aspx) within a transaction.

As a result, because of the Windows OS [Transactional NTFS](http://msdn.microsoft.com/en-us/library/bb968806(VS.85).aspx), the back requirement and the implementation of operating sequences back in case of abnormalities of the “5.4.1.11” requirement are realized.

## Addressing 5.4.1.12 “The OS shall provide a utility program for inspecting the integrity of file systems and disks and the utility program can be operated automatically by the OS”

The Windows OS addresses the “5.4.1.12” requirement because of the availability of the “[chkdsk command line tool](http://technet.microsoft.com/en-us/library/cc730714.aspx)” as described in the “Self healing Windows OS NTFS” section under Appendix B of this paper.

Due to the default inclusion of the text string “autocheck autochk \*” in the “BootExecute” registry key value under the “HKLM\SYSTEM\CurrentControlSet\Control\Session Manager” registry key, the Windows OS Session Manager (SMSS.exe) launches the “autochk.exe” program automatically. The Windows OS Session Manager (SMSS.exe) is the first user mode Windows OS process to start in the Windows OS boot sequence. The “autochk.exe” program detects the all mounted NTFS partitions/volumes for the dirty bit (VOLUME\_IS\_DIRTY) using the [FSCTL\_IS\_VOLUME\_DIRTY](http://msdn.microsoft.com/en-us/library/ms794373.aspx). The “autochk.exe” program launches the “[chkdsk command line tool](http://technet.microsoft.com/en-us/library/cc730714.aspx)” for a specific NTFS partition/volume after it detects that the NTFS partition/volume is dirty. The Windows OS Session Manager only sets the “\Device\VolumesSafeForWriteAccess” Windows OS event after the “autochk.exe” program and the chkdsk tool have indicated success. The chkdsk tool produces its log messages in the “BOOTEX.LOG” file, which resides in the root of the specific NTFS partition/volume.

As described in the “Self healing Windows OS NTFS” section under Appendix B of this paper, the Windows OS NTFS marks a specific NTFS partition/volume dirty after it detects a corruption and its self healing capability is not able to repair the corruption.

## Addressing 5.4.1.13 “In case of the failure or interruption of the OS, a restoration of the OS with the minimum amount of damage is available”

The “5.4.1.13” requirement additionally requires that “in case of the interruption of the OS because of malfunction or other reasons, a mechanism is available to restore the OS. The OS has the running capability under its maintenance mode, which can only be used by an administrator, when the restoration occurs”.

The Windows OS addresses the “5.4.1.13” requirement. Following the capabilities and policies explained in the “Windows OS Backup and Restore” section under Appendix B of this paper, the administrators have the necessary Windows OS tools to backup the system state of a Windows OS machine into a specific backup version after the machine has been properly configured. When required, the administrators deploy the capabilities, as explained in the “Characterizing the Windows OS “Maintenance Modes”” section under Appendix B of this paper, to restore the Windows OS from a desired backup version.

## Addressing 5.4.1.14 “The OS shall provide the processes for software and data backup and restoration, where a synchronization point for restart is added to a backup instance for a subsequent restoration”

The Windows OS addresses the “5.4.1.14” requirement. The Windows OS Backup and Restore capability supports the concept of backup versioning. As described in the “Windows OS Backup and Restore” section under Appendix B of this paper, there are the “[Wbadmin get versions](http://technet.microsoft.com/en-us/library/cc742116.aspx)” and the “[Wbadmin get items](http://technet.microsoft.com/en-us/library/cc742041.aspx)” commands for querying specific backup versions and their corresponding backup items. The concept of backup versioning identifies a specific backup instance which has been made at a specific point of the past in the “MM/DD/YYYY-HH:MM” format. This specific point of the past is therefore deemed as a synchronization point for restarting during a restore operation attempt.

# Meeting the “Information Protection Persistent Storage Data Availability Management Requirements”

There is no individual functional requirement under the heading of “Information Protection Persistent Storage Data Availability Management Requirements”.

# Meeting the “Information Protection Persistent Storage Data Availability Audit Requirements”

There is 1 individual functional requirement under the heading of “Information Protection Persistent Storage Data Availability Audit Requirements”. It is listed as “5.4.3.1”.

## Addressing 5.4.3.1 “To the data of the OS stored on disk, the OS shall realize an “automatically record problems of disk surface” function”

The Windows OS addresses the “5.4.3.1” requirement as follows.

Generating an audit record for (i.e. recording) problems of disk surface may not be too useful the user to recover from a corruption due to disk surface problems. Instead, as described in the “Self healing Windows OS NTFS” section under Appendix B of this paper, the Windows OS NTFS provides extensive information within the “RepairDetail” field when it generates the following two soft audit event records:

* Event ID 130 (NTFS\_ETW\_REPAIR\_SUCCESS) “The file system structure on volume <Volume ID> has now been repaired”;
* Event ID 131 (NTFS\_ETW\_REPAIR\_FAILED) “The file system structure on volume <Volume ID> cannot be corrected. Please run the chkdsk utility on the volume <Volume ID>”.

# Meeting the “Information Protection Maintenance Mode Access Control Functional Requirements”

There are 5 individual functional requirements under the heading of “Information Protection Maintenance Mode Access Control Functional Requirements”. They are listed as “5.5.1.n”, where n = 1, 2, 3, 4, and 5.

## Addressing 5.5.1.1 “The ordinary operation mode and system maintenance mode of the OS should be differentiated”

The Windows OS addresses the “5.5.1.1” requirement because the process of an administrative recovery, from the various maintenance modes characterized in the Windows OS, begins with either a proper user authentication or the necessary check for the possession of a specific secret (such as the private key associated with the EFS Recovery Agent certificate public key or the password key disk for a user account with a forgotten password), as explained in the following “Characterizing the Windows OS “Maintenance Modes”” section.

### Characterizing the Windows OS “Maintenance Modes”

Due to the availability of several protection levels in the Windows OS’s “defense in depth” concept, there is not a single “maintenance mode” within the Windows OS. As the security mechanism(s) are different at different protection levels, the recovery tools of the corresponding maintenance modes are also necessarily different. In this paper, we characterize a maintenance mode at a protection level by the availability of [specific Windows OS recovery tools](http://technet.microsoft.com/en-us/library/cc162802.aspx) for an authorized subject’s recovery from a Windows OS mechanism failure at the protection level.

#### BitLocker™ Full volume encryption maintenance mode

In the case where the BitLocker™ Full volume encryption is enforced, its maintenance mode is characterized by the use of the following four recovery tools:

* The [BitLocker™ recovery console](http://technet.microsoft.com/en-us/library/cc162802.aspx), which runs before Windows Vista™ boots, is designed to help users unlock a BitLocker™-encrypted operating system volume
  + It is part of the Windows OS boot manager;
  + The BitLocker™ recovery console is entered by pressing the “ESC” key after the appearance of the following “fve-bad-pin” message as described in the “TPM unsealing during volume open” section of this paper:
    - “That PIN is incorrect. If you entered the PIN with the number keys, use the function keys F1 - F10 instead. Press ENTER to try another PIN. Press ESC for recovery”;
* The [BitLocker™ control panel’s recovery wizard](http://technet.microsoft.com/en-us/library/cc162802.aspx) (FVERecover.dll) is designed to help users unlock a BitLocker™-encrypted data volume (namely, a non-operating system volume, an alternate operating system volume on the same computer, or an operating system volume from another computer)
  + It runs in the security context of the caller subject;
* The [Windows Vista Recovery Environment (WinRE)](http://technet.microsoft.com/en-us/library/cc749147.aspx) includes a wizard (RecEnv.exe) that can be used to unlock BitLocker™-protected operating system or data volumes
  + WinRE is available from a Windows Vista DVD or within a recovery (hidden) partition available from some computer manufacturers;
* The [BitLocker™ Repair Tool](http://support.microsoft.com/kb/928201) (repair-bde.exe) helps access encrypted data if the drive/volume has been severely damaged
  + It is used in the situation where the [BitLocker™ recovery console](http://technet.microsoft.com/en-us/library/cc162802.aspx) cannot be started in the Windows OS boot manager’s operational environment.

All the above recovery tools for the BitLocker™ Full volume encryption maintenance mode require a valid [BitLocker™ recovery password](http://technet.microsoft.com/en-us/library/cc766295.aspx) for the drive/volume being recovered. Following [the Microsoft deployment guide for BitLocker™](http://technet.microsoft.com/en-us/library/cc162802.aspx), it is critical that [BitLocker™ recovery passwords](http://technet.microsoft.com/en-us/library/cc766295.aspx) are possessed by the administrators only and copies of the [BitLocker™ recovery passwords](http://technet.microsoft.com/en-us/library/cc766295.aspx) are stored in safe places where only the administrators have the authorized access. The storage media for [BitLocker™ recovery passwords](http://technet.microsoft.com/en-us/library/cc766295.aspx) may be:

* [A USB drive](http://technet.microsoft.com/en-us/library/cc766295.aspx);
* [A network drive](http://technet.microsoft.com/en-us/library/cc766295.aspx);
* [A sheet of printed paper](http://technet.microsoft.com/en-us/library/cc766295.aspx);
* [The Windows OS Active Directory](http://technet.microsoft.com/en-us/library/cc766015.aspx).

As a result, the BitLocker™ Full volume encryption maintenance mode addresses the “5.5.1.2” requirement that “The OS shall prevent an ordinary user from entering the maintenance mode from unauthorized system, and prevent the ordinary user from interacting with maintenance mode of the system, thus guaranteeing that the installation and configuration of the system are implemented in a safe mode before the access of ordinary users”.

#### Post BitLocker™ Full volume encryption maintenance mode

In the case where the BitLocker™ Full volume encryption is not configured, a [recovery may still be necessary when the Windows OS fails to boot](http://www.microsoft.com/whdc/system/platform/firmware/OEMBoot_Vista.mspx). The post BitLocker™ Full volume encryption maintenance mode is characterized by the use of the following [Windows Recovery Environment (WinRE)](http://technet.microsoft.com/en-us/library/cc749147.aspx) recovery tool.

* The [Windows Recovery Environment (WinRE)](http://technet.microsoft.com/en-us/library/cc749147.aspx) includes a wizard (RecEnv.exe) that can be used to unlock BitLocker™-protected operating system or data volumes
  + WinRE is available from a Windows Vista DVD or within a recovery partition available from some computer manufacturers.

If set up according to [the Windows RE Image deployment walkthrough](http://technet.microsoft.com/en-us/library/cc721939.aspx), the RecEnv.exe wizard and its “WinREConfig.xml” configuration file and certain supporting tools reside in a [hidden (type 0x27) partition/volume](http://download.microsoft.com/download/9/c/5/9c5b2167-8017-4bae-9fde-d599bac8184a/Boot_Modifications.doc).

After the RecEnv.exe wizard is launched from its hidden partition/volume due to a Windows OS boot failure, by default, it requires a user authentication when the following conditions are met:

* The location of the Windows OS being recovered is not external;
* The Windows OS SAM username/password database associated with the Windows OS being recovered is neither corrupted nor inaccessible.

The user authentication is based on the user name and password combination only, defined in the local Windows OS SAM username/password database. An administrator can configure the “[AlwaysAuthenticate](http://technet.microsoft.com/en-us/library/cc749546.aspx)” option in the “WinREConfig.xml” file. If the “[AlwaysAuthenticate](http://technet.microsoft.com/en-us/library/cc749546.aspx)” option is configured, then the user authentication is always enforced within the RecEnv.exe wizard, regardless of the location of the Windows OS belong recovered or the accessibility of the Windows OS SAM username/password database associated with the Windows OS being recovered. The Windows OS SAM username/password database resides within the “HKEY\_LOCAL\_MACHINE\SAM” registry key of the Windows OS registry. If the user authentication fails, then the following message is displayed to the user before the RecEnv.exe wizard exits.

* “You must log in to access System Recovery Options. If you are having trouble logging in, please contact your computer administrator for assistance. Click OK to restart the computer”.

We note that the RecEnv.exe wizard is incapable to collect from the user the [system key (syskey) policy](http://support.microsoft.com/kb/818200) master key if the master key need to be supplied by the user due to the configured [system key (syskey) policy](http://support.microsoft.com/kb/818200). If the Windows OS SAM username/password database associated with the Windows OS being recovered is encrypted [according to the system key (syskey) policy](http://support.microsoft.com/kb/818200) with a user supplied master key, then the Windows OS SAM username/password database is deemed as inaccessible.

We also note that the RecEnv.exe wizard allows only a maximum of 7 attempts for conducting the password based user authentication with a password supplied by the local user. After 7 failed user authentication attempts, the following message is displayed to the user before the RecEnv.exe wizard exits.

* “You are having trouble logging in. Please contact your computer administrator for assistance”.

Assuming that the RecEnv.exe wizard successfully authenticates the user, the RecEnv.exe wizard displays a list of supported recovery tools available to the authenticated user, based on the authenticated user’s membership in the “Administrators” local group.

By default, the following list of recovery tools is displayed to the authenticated user who is not a member of the “Administrators” local group:

* The “Startup Repair” tool (StartRep.exe)
  + Its purpose is to “Automatically (attempt to) fix problems that are preventing Windows from starting”;
  + It is available in a Windows Vista machine only.

By default, the following list of recovery tools is displayed to the authenticated user who is a member of the “Administrators” local group:

* The “System Restore” tool (Rstrui.exe)
  + Its purpose is to “Restore Windows to an earlier point in time”;
  + It is available in a Windows Vista machine only.
* The “Windows Complete PC Restore” tool (bmrui.exe)
  + Its purpose is to “Restore the entire Windows server or personal computer from a backup image”;
  + It is available in a Windows Vista or Windows Server 2008 machine only.
* The “Windows Memory Diagnostic Scheduler” tool (mdsched.exe)
  + Its purpose is to “Check the Windows computer for memory hardware errors by using the “memtest.exe” or “memtest.efi” memory testing application”;
  + It is available in a Windows Vista or Windows Server 2008 machine only.
* The “Command Prompt” tool (cmd.exe)
  + Its purpose is to “Open a command prompt window”;
  + It is available in a Windows Vista or Windows Server 2008 machine only.

Similarly, the lists of recovery tools available to a standard user and to an administrator can be configured through the “[NonAdminToolsOnly](http://technet.microsoft.com/en-us/library/cc749546.aspx)”, “[RecoveryTools](http://technet.microsoft.com/en-us/library/cc749546.aspx)” and “[SupportTool](http://technet.microsoft.com/en-us/library/cc749546.aspx)” options in the “WinREConfig.xml” file, which resides in an administrator defined hidden partition/volume with the RecEnv.exe wizard.

As a result, the post BitLocker™ Full volume encryption maintenance mode addresses the “5.5.1.2” requirement that “The OS shall prevent an ordinary user from entering the maintenance mode from unauthorized system, and prevent the ordinary user from interacting with maintenance mode of the system, thus guaranteeing that the installation and configuration of the system are implemented in a safe mode before the access of ordinary users”.

##### The Specific memory tests conducted by the Microsoft Memory Testing Application

The Microsoft “memtest.exe” or “memtest.efi” memory testing application conducts the following memory tests.

* The “Basic” tests:
  + MATS+
    - It identifies most basic faults using walking ones and zeroes;
  + INVC
    - It performs one and zero fills in order to locate inverse coupling faults;
  + SCHCKR (cache enabled)
    - It applies a simple checkerboard pattern.
* The “Standard” tests (default):
  + The “Basic” tests (described in the above);
  + LRAND
    - It applies random data to all of memory;
  + Stride6 (cache enabled)
    - It applies a stride-6 pattern using data patterns tailored for the memory in the underlying system;
  + CHCKR3
    - It applies a 3x3 extended checkerboard pattern;
  + WMATS+
    - It runs a normal MATS+ test using data patterns tailored for the memory in the underlying system;
  + WINVC
    - It runs an inverse coupling test using data patterns tailored for the memory in the underlying system.
* The “Extended” tests:
  + The “Basic” tests (described in the above);
  + The “Standard” tests (described in the above);
  + MATS+ (cache disabled)
    - It identifies most basic faults using walking ones and zeroes;
  + Stride38
    - It uses a stride-38 pattern to simulate a checkerboard of ones and zeroes;
  + WSCHCKR
    - It runs a simple checkerboard test using data patterns tailored for the memory in the underlying system;
  + WStride-6
    - It applies a stride-6 pattern using data patterns tailored for the memory in the underlying system;
  + CHCKR4
    - It applies a 4x4 extended checkerboard pattern;
  + WCHCKR3
    - It applies a 3x3 checkerboard using data patterns tailored for the memory in the underlying system;
  + ERAND
    - It makes several passes through memory applying random data patterns;
  + Stride6 (cache disabled)
    - It uses a stride-6 pattern to simulate a checkerboard of ones and zeroes;
  + CHCKR8
    - It applies an 8x8 extended checkerboard pattern.

#### “Crash on Audit Fail” maintenance mode

If the “CrashOnAuditFail” policy is set, then the Windows OS audit policy unit crashes the Windows OS machine to produce [STATUS\_AUDIT\_FAILED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) (0xC0000244L) on the blue screen when it is unable to trace (i.e. write) a generated audit event record through the [NtTraceEvent()](http://msdn.microsoft.com/en-us/library/aa364137.aspx) interface of the [Windows OS event tracing (ETW) facility](http://msdn.microsoft.com/en-us/library/aa363668(VS.85).aspx) with the “ETW\_NT\_FLAGS\_TRACE\_SECURITY” flag. A full security log file (i.e. its maximum size has been reached) always causes the Windows OS audit policy unit inability to trace (i.e. write) a generated audit event record. We therefore characterize the blue screen displaying the [STATUS\_AUDIT\_FAILED](http://msdn.microsoft.com/en-us/library/cc704588.aspx) (0xC0000244L) as the “CrashOnAuditFail” maintenance mode. After the user physically pushes the power button of the machine, following the appearance of the “CrashOnAuditFail” maintenance mode, the “Request logon credentials” state of the first instance of the window logon state maintaining service eventually is entered. The subsequent states led by this “Request logon credentials” state allow only an administrator to logon. Other users are denied to logon, until a successfully logged on administrator has successfully cleared the security log file and reconfigured the “CrashOnAuditFail” policy, and thereby has recovered the Windows OS machine from the “CrashOnAuditFail” maintenance mode.

As a result, the “CrashOnAuditFail” maintenance mode addresses the “5.5.1.2” requirement that “The OS shall prevent an ordinary user from entering the maintenance mode from unauthorized system, and prevent the ordinary user from interacting with maintenance mode of the system, thus guaranteeing that the installation and configuration of the system are implemented in a safe mode before the access of ordinary users”.

#### Recovery from Encrypting File System (EFS) protection maintenance mode

In the case where the encrypting file system (EFS) is enforced for specific user (non system) files or file directories, its maintenance mode is characterized by the intervention of an authorized EFS Recovery Agent in the following manner.

Recall from the “Addressing 6.1.1.1 “The OS shall ensure that security attributes on named objects, when exported to removable media, are associated with the object”” section of this paper, an EFS-encrypted file object could include a [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) in its “$EFS” metadata stream, depending on the Windows OS domain wide [EFS policy](http://msdn.microsoft.com/en-us/library/cc232262.aspx). The [Data Recovery Field](http://technet.microsoft.com/en-us/library/cc781588.aspx) contains the encrypted File Encryption Key (FEK). The FEK is RSA-encrypted with the EFS Recovery Agent certificate public key of an EFS Recovery Agent specified in the [EFS policy](http://msdn.microsoft.com/en-us/library/cc232262.aspx). An authorized EFS Recovery Agent user is a user, whose user account possesses the private key associated with the EFS Recovery Agent certificate public key. By default, authorized EFS Recovery Agent users are administrators.

For an authorized EFS Recovery Agent user to recover an EFS-encrypted file object, the EFS Recovery Agent user logs onto the Windows OS machine, where the EFS-encrypted file object resides in. In a “Windows Explorer” window or a “Command Prompt” window, the EFS Recovery Agent user goes to the file system location, where the EFS-encrypted file object resides in. The EFS Recovery Agent user either

* unchecks the “Encrypt contents to secure data” box in the “Advanced Attributes” property dialog (i.e. the EFS security attribute) associated with the EFS-encrypted file object in the “Windows Explorer” window;
* enters the “[Cipher.exe /D <the name of the EFS-encrypted file object>](http://technet.microsoft.com/en-us/library/cc771346.aspx)” command in the “Command Prompt” window.

Because of the EFS Recovery Agent user’s possession of the private key associated with the EFS Recovery Agent certificate public key, the result of either of the above two actions is a successful decryption of the EFS-encrypted file object. As a result, the clear text of the EFS-encrypted file object is recovered and the EFS Recovery Agent user may then log off from the Windows OS machine. This completes the characterization of the “Recovery from Encrypting File System (EFS) protection” maintenance mode.

Therefore, the “Recover from Encrypting File System (EFS) protection” maintenance mode addresses the “5.5.1.2” requirement that “The OS shall prevent an ordinary user from entering the maintenance mode from unauthorized system, and prevent the ordinary user from interacting with maintenance mode of the system, thus guaranteeing that the installation and configuration of the system are implemented in a safe mode before the access of ordinary users”.

#### Recovering user password maintenance mode

In the case where the recovery of the current user password for a local Windows Vista user account is necessary, its maintenance mode is characterized by the intervention of an authorized user physically possessing the user account’s password key disk according to the user manual of the “[Forgotten Password Wizard](http://support.microsoft.com/kb/930381)” in the following manner.

Recall from the “NTLM security provider notifies the local Windows OS data protection API manger” section of this paper, the current user password is encrypted with the user account’s password recovery 2048-bit public key. The password recovery 2048-bit public key and its corresponding private key would have been generated previously for the user account when the account’s owner exercised the “[Forgotten Password Wizard](http://support.microsoft.com/kb/930381)”. The password recovery private key would also have been saved to a removable password key disk according to the user manual of the “[Forgotten Password Wizard](http://support.microsoft.com/kb/930381)”.

To use the password key disk to recover the situation where the user has forgotten the user password for the local Windows Vista user account, the user, possessing the password key disk, follows the user manual of the “[Reset Password Wizard](http://support.microsoft.com/kb/930381)” as follows.

1. The user clicks the “Reset Password…” button in the logon dialog of the Window logon UI service after the Window Logon State Maintaining Service receives a logon failure from the Windows OS Authentication Service due to the user’s previously entered wrong password.
2. The user puts the password key disk in an applicable media drive, and then clicks “Next” in the “[Reset Password Wizard](http://support.microsoft.com/kb/930381)”.
3. In the “The password key disk is in the following drive list”, the users clicks the drive in which the password reset disk is located, and then clicks “Next”.
4. In the “Type a new password” box, the user types the new password that the user wants to use for the user account.
5. In the “Type the password again to confirm” box, the user types the password again.
6. The user then clicks “Next”, and then clicks “Finish”.

The result of the above “[Reset Password Wizard](http://support.microsoft.com/kb/930381)” actions is a successful reset of the user password due to the availability of the password key disk.

As a result, the “Recovering user password” maintenance mode addresses the “5.5.1.2” requirement that “The OS shall prevent an ordinary user from entering the maintenance mode from unauthorized system, and prevent the ordinary user from interacting with maintenance mode of the system, thus guaranteeing that the installation and configuration of the system are implemented in a safe mode before the access of ordinary users”.

## Addressing 5.5.1.2 “The OS shall prevent an ordinary user from entering the maintenance mode from unauthorized system”

The “5.5.1.2” requirement additionally requires that “The OS shall prevent the ordinary user from interacting with maintenance mode of the system, thus guaranteeing that the installation and configuration of the system are implemented in a safe environment before the access of ordinary users”.

The “5.5.1.2” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “1.1.1.2” requirement “The OS shall ensure that all security policies are enforced before each security function is allowed to proceed”.

The Windows OS addresses the “5.5.1.2” requirement because an administrative recovery, from the various maintenance modes characterized in the Windows OS, requires a proper user authentication, as explained in the “Characterizing the Windows OS “Maintenance Modes”” section under Appendix B of this paper”.

We also note that a managed Windows OS machine within a Windows OS domain/forest automatically receives and processes centrally distributed system and security configuration policies through its Windows OS group policy client service (gpsvc.dll) after it successfully starts up, as explained in the “Group policy processing at a local computer” section of this paper. Therefore, the desired Windows OS domain-wise system and security configuration policies, as specified by the authorized administrators, would have been enforced within the managed Windows OS machine before any user is attempted to logon to the machine.

## Addressing 5.5.1.3 “For backup or conventional system maintenance tasks which do not affect the security subsystem of the OS, not all these system maintenance tasks are carried out under the maintenance mode”

The Windows OS addresses the “5.5.1.3” requirement because the Windows OS backup and restore is based on the Windows OS Volume Shadow Copy Service (VSSVC.exe) and the Windows OS Block Level Backup Engine Service (wbengine.exe). As explained in the “Windows OS Backup and Restore” section of this paper, the Windows OS services and applications do not need to be interrupted when their data need to be copied for backup or restore. An explicit entry into a maintenance mode is not necessary.

We also note that a managed Windows OS machine within a Windows OS domain/forest automatically receives and processes centrally distributed system and security configuration policies through its Windows OS group policy client service (gpsvc.dll) after it successfully starts up, as explained in the “Group policy processing at a local computer” section of this paper. Therefore, the corresponding system maintenance tasks occur automatically. An explicit entry into a maintenance mode is also not necessary.

## Addressing 5.5.1.4 “After the completion of the OS installation and before the access by ordinary users, the OS shall configure responsibilities of initial users and administrators, root directory, audit parameters, system audit trail setup, and appropriate access control on files and directories”

The “5.5.1.4” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “1.2.2.1” requirement “The OS shall provide the ability for an authorized administrator to remotely manage the OS”.

“The Windows OS also addresses the “5.5.1.4” requirement. [The group policy objects processed by the Windows OS group policy client service (gpsvc.dll)](http://www.microsoft.com/downloads/details.aspx?FamilyID=41dc179b-3328-4350-ade1-c0d9289f09ef&DisplayLang=en) include the configuration aspects specified in the “5.5.1.4” requirement.

Please see the justification text for addressing the Commercial Grade OS Requirement Set “1.2.2.1” requirement for detail.

## Addressing 5.5.1.5 “The OS shall have the ability to run under the maintenance mode, and the OS can only be used by system administrators under the maintenance mode when specific security functions fail”

The “5.5.1.5” requirement treats similar security concern(s) as the removed Commercial Grade OS Requirement Set “5.3.1.2” requirement “After a failure of a self test, the OS shall ensure a maintenance mode where the ability to return the OS to a secure state is provided”.

The Windows OS equally does not meet the “5.5.1.5” requirement as the Windows OS does not incorporate self tests for its security functions in order to determine whether all its security functions have failed.

However, as parts of its serviceability and sustained engineering processes, Microsoft continues to conduct unit tests and scenario based tests for the Windows OS after the Windows OS is shipped to customers. Issues arisen during the serviceability and sustained engineering processes are investigated and addressed in the next scheduled service pack of the Windows OS.

# Meeting the “Information Protection Maintenance Mode Access Control Management Requirements”

There is no individual management requirement under the heading of “Information Protection Maintenance Mode Access Control Management Requirements”.

# Meeting the “Information Protection Maintenance Mode Access Control Audit Requirements”

There no individual audit requirement under the heading of “Information Protection Maintenance Mode Access Control Audit Requirements”.

# Meeting Additional “Import/Export of Data Exported Data Functional Requirements”

The Commercial Grade OS Requirement Set already has 1 individual functional requirement under the heading of “Import/Export of Data Exported Data Functional Requirements”. It is listed as “6.1.1.1”.

There is 1 additional individual functional requirement under the heading of “Import/Export of Data Exported Data Functional Requirements” in this appendix. It is listed as “6.1.1.2”.

## Addressing 6.1.1.2 “When information is sent from the scope under the OS control to the scope beyond the OS control, sensitive labels can be attached or not attached”

The Windows OS does not meet the “6.1.1.2” requirement.

The Windows OS also does not meet the counterpart requirement for the integrity labels either.

When the information belonging to a file is exported, the Windows OS does not attach the file’s sensitivity label/level or integrity label/level with the information. However, the Windows OS could encrypt the information being exported with the Windows OS encrypting file system (EFS) service, where the EFS Data Decryption Field and Data Recovery Field are the associated security attributes. Please see the following section of this paper for detail.

* “Addressing 6.1.1.1 “The OS shall ensure that security attributes on named objects, when exported to removable media, are associated with the object”.

# Meeting Additional “Import/Export of Data Imported Data Functional Requirements”

The Commercial Grade OS Requirement Set already has 1 individual functional requirement under the heading of “Import/Export of Data Imported Data Functional Requirements”. It is listed as “6.2.1.1”.

There is 1 additional individual functional requirement under the heading of “Import/Export of Data Imported Data Functional Requirements” in this appendix. It is listed as “6.2.1.2”.

## Addressing 6.2.1.2 “When information is sent from the scope beyond the OS control to the scope under the OS control, sensitive labels should be attached”

The Windows OS does not meet the “6.2.1.2” requirement.

The Windows OS also does not meet the counterpart requirement for the integrity labels either.

When the information belonging to a file is imported by an importer subject, the Windows OS does not attach the file’s sensitivity label/level or integrity label/level with the information.

However, the information belonging to the file being imported is subject to

* the “no write down” rule of the “Poor man’s Mandatory Access Control (MAC)”;
* the Mandatory Integrity Control Policy,

as explained in the following.

The information belonging to the file being imported need to reside in

* a new file of the local Windows OS file system;
* an existing destination file of the local Windows OS file system as updated data.

In the case of importing to a new file, the creation of the new file occurs automatically, but the creation is subject to

* the “no write down” rule of the “Poor man’s Mandatory Access Control (MAC)” according to the sensitivity level (SL) of the importer subject and the sensitivity level (SL) of the destination file directory for the new file;
* the Mandatory Integrity Control Policy, where
  + the Mandatory Integrity Control Policy “SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP” rule, as stated in the justification text for addressing the Commercial Grade OS Requirement Set “2.2.1.5” requirement, according to the integrity level (IL) of the importer subject and the integrity level (IL) of the destination file directory for the new file; or
  + the new file receives the default Mandatory Integrity Control Policy SYSTEM\_MANDATORY\_LABEL\_ACE ACE of
    - an integrity level value of MandatoryLevelMedium (2) (SepDefaultMandatorySid);
    - the write (SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP) access operation class.

In the case of importing to an existing destination file, the writing to the existing destination file occurs automatically, but the writing is subject to

* the “no write down” rule of the “Poor man’s Mandatory Access Control (MAC)” according to the sensitivity level (SL) of the importer subject and the sensitivity level (SL) of the existing destination file;
* the Mandatory Integrity Control Policy “SYSTEM\_MANDATORY\_LABEL\_NO\_WRITE\_UP” rule, as stated in the justification text for addressing the Commercial Grade OS Requirement Set “2.2.1.5” requirement, according to the integrity level (IL) of the importer subject and the integrity level (IL) of the existing destination file.

For the application of the Mandatory Integrity Control Policy when importing user data, please see the following section of this paper for detail.

* Addressing 6.2.1.1 “The OS shall enforce a specific set of rules when importing user data”.

# Meeting Additional “Auditing Audit Collection Functional Requirements”

The Commercial Grade OS Requirement Set already has 4 individual functional requirements under the heading of “Auditing Audit Collection Functional Requirements”. They are listed as “8.1.1.n”, where n = 1, 2, 3 and 4.

There are 6 additional individual functional requirements under the heading of “Auditing Audit Collection Functional Requirements” in this appendix. They are listed as “8.1.1.n”, where n = 5, 6, 7, 8, 9, and 10.

## Addressing 8.1.1.5 “Security audit function shall be closely combined with authentication, discretionary access control, labeling, mandatory access control, integrity control and other security functions”

The “8.1.1.5” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.1.1.1” requirement “The OS shall be able to generate audit records for all security-relevant events identified in this Commercial Grade OS Requirement Set and the other specific security relevant auditable events designed to be generated by the OS claiming compliance with this Commercial Grade OS Requirement Set”.

The Windows OS also addresses the “8.1.1.5” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.1.1.1” requirement for detail.

## Addressing 8.1.1.6 “The OS shall point out the least types of recordable audit events, including the change of system database management -- the addition or change of system procedures or processes”

The Windows OS addresses the “8.1.1.6” requirement as the Windows OS typically stores its system configuration data in the Windows OS registry or the Windows OS Active Directory (AD). After an authorized subject is permitted to make a change to a Windows OS registry key or a Windows OS AD object, the following audit records are generated respectively”:

* [Event ID 4657](http://support.microsoft.com/kb/947226) “A registry value was modified” for success with the information about the change;
* [Event ID 5136](http://support.microsoft.com/kb/947226) “A directory service object was modified” for success with the information about the change.

We recommend an explicit description of the security relevant system procedure or process changes that shall cause the generation of a corresponding audit record. The explicit description helps the developers of the compliant OS products to understand the precise intent about the required audit generation.

## Addressing 8.1.1.7 “The OS shall point out the least types of recordable audit events, including the change of system database management -- the change of date and time”

We deem that the local machine time is changed when there is a difference between the local machine time and the time from an authoritative time source and the difference is big enough so that the Windows OS [Time service](http://technet.microsoft.com/en-us/library/cc756502.aspx) is not able to resolve the difference automatically, according to its configuration policies.

The Windows OS addresses the “8.1.1.7” requirement because of the Windows [Time service’s](http://technet.microsoft.com/en-us/library/cc756502.aspx) generation of the [Event ID 34](http://technet.microsoft.com/en-us/library/cc733187.aspx) (MSG\_TIME\_CHANGE\_TOO\_BIG) soft audit record as explained in the following.

The Windows OS [Time service](http://technet.microsoft.com/en-us/library/cc756502.aspx) (W32time.dll) synchronizes local Windows OS machine’s time with a time source. The Windows OS Time service on a Windows OS domain controller can be configured as either a reliable or an unreliable time source. The Windows Time service running on a client Windows OS machine attempts to synchronize its time source with servers that are indicated as being reliable. The Windows OS Time service can be configured to recognize a specific Windows OS domain controller within its Windows OS domain as a reliable time source, and it synchronizes itself periodically with this source.

The Windows Time service internal time synchronization process involves the following steps:

* Input providers request and receive time samples from configured [RFC 1305 Network Time Protocol (NTP)](http://www.ietf.org/rfc/rfc1305.pdf) time sources;
* These time samples are then passed to the Windows Time Service Manager, which collects all the samples and passes them to the clock discipline subcomponent;
* The clock discipline subcomponent applies all NTP algorithms and selects the best time sample;
* The clock discipline subcomponent adjusts the time of the local Windows OS machine’s system clock to the most accurate time by either adjusting the clock rate or directly changing the time.

The Windows Time service generates the following soft audit event record to the soft audit storage[[39]](#footnote-40)

* [Event ID 34](http://technet.microsoft.com/en-us/library/cc733187.aspx) (MSG\_TIME\_CHANGE\_TOO\_BIG) “The time service has detected that the system time needs to be changed by <N> seconds. The time service will not change the system time by more than <M> seconds. Verify that your time and time zone are correct, and that the time source <the name of the time source server for the next time synchronization> is working properly”, where
  + the value N is the data (in milliseconds) of the “[LargePhaseOffset](http://technet.microsoft.com/en-us/library/cc758543.aspx)” registry key value under the following registry key for indicating that the Windows Time service recognizes an error when the time difference between the local time and the time source’s time is greater than or equal to this “[LargePhaseOffset](http://technet.microsoft.com/en-us/library/cc758543.aspx)” registry key value data
    - HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Config;
  + the value M is the data (in seconds) of the “[SpikeWatchPeriod](http://technet.microsoft.com/en-us/library/cc787657.aspx)” under registry key value under the following registry key and it specifies the amount of time, in seconds, that a suspicious offset must persist before it is accepted as correct
    - HKLM\SYSTEM\CurrentControlSet\Services\W32Time\Config.

The default of the “[LargePhaseOffset](http://technet.microsoft.com/en-us/library/cc758543.aspx)” registry key value data is 50000000.

The default of the “[SpikeWatchPeriod](http://technet.microsoft.com/en-us/library/cc787657.aspx)” registry key value data is 900.

## Addressing 8.1.1.8 “Data records of each (audit) event shall include specific information”

The “8.1.1.8” requirement requires the following specific information:

* the date and time of events, the user of trigger events, event types, and the success or failure of events;
* to identification and authorization, request information should be recorded (such as terminal or network address);
* to events on creating and deleting, names and security attributes of objects should be recorded.

Except the “user of trigger events” aspect, the “8.1.1.8” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.1.1.4” requirement “The OS shall record, within each audit record, the specific set of information items, as appropriate to the audit event”. The Commercial Grade OS Requirement Set requires the following set of information items to be recorded within an audit event record, as appropriate to the corresponding audit event:

1. date, time, and location of the event;
2. type of event;
3. event outcome (success or failure);
4. name of the object;
5. old and new values (except for authentication data and critical cryptographic security parameters) of changed security relevant data.

Except the “user of trigger events” aspect, the Windows OS also addresses the “8.1.1.8” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.1.1.4” requirement for detail.

For the “user of trigger events” aspect, the “8.1.1.8” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.1.1.2” requirement “The OS shall be able to associate each auditable event with the identity or the user that caused the event”.

For the “user of trigger events” aspect, the Windows OS also addresses the “8.1.1.8” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.1.1.2” requirement for detail.

## Addressing 8.1.1.9 “The OS shall provide audit logs, real-time alarm generation, potential security risk analysis, basic abnormality inspection”

The “8.1.1.9” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.1.1.3” requirement “The OS shall be able to monitor and report the accumulation of specific sets of audit events known to indicate a potential security violation and immediately report the accumulated events when a threshold is exceeded”.

The Windows OS also addresses the “8.1.1.9” requirement with the capabilities provided by [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx). Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.1.1.3” requirement for detail.

## Addressing 8.1.1.10 “The OS shall provide the history records on modification of system resources that can be accessed by users”

The “8.1.1.10” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.1.1.4” requirement “The OS shall record, within each audit record, the specific set of information items, as appropriate to the audit event”. The Commercial Grade OS Requirement Set requires the following set of information items to be recorded within an audit event record, as appropriate to the corresponding audit event:

1. date, time, and location of the event;
2. type of event;
3. event outcome (success or failure);
4. name of the object;
5. old and new values (except for authentication data and critical cryptographic security parameters) of changed security relevant data.

The Windows OS also addresses the “8.1.1.10” requirement for the modification of security relevant system resources. As a modification of a specific security relevant system resource occurs, the corresponding event audit record (for success) is generated. This (success) audit record indicates the corresponding value(s) of the specific security relevant system resource. As a result, the indicated value(s) in the audit records during a time period establish the history records for security relevant system resource modification in the same time period.

Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.1.1.4” requirement for detail.

# Meeting Additional “Auditing Audit Collection Management Requirements”

The Commercial Grade OS Requirement Set already has 2 individual management requirements under the heading of “Auditing Audit Collection Management Requirements”. They are listed as “8.1.2.n”, where n = 1, and 2.

There are 5 additional individual functional requirements under the heading of “Auditing Audit Collection Management Requirements” in this appendix. They are listed as “8.1.2.n”, where n = 3, 4, 5, 6, and 7.

## Addressing 8.1.2.3 “The OS shall provide security audit event selection”

The “8.1.2.3” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.1.2.2” requirement “The OS shall allow an authorized administrator to specify which events, from the set of auditable events, are to be audited”.

The Windows OS also addresses the “8.1.2.3” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.1.2.2” requirement for detail.

## Addressing 8.1.2.4 “The OS shall point out the least types of recordable audit events, including the change of system database management -- audit trail setup and analysis”

We deem the act of an authorized administrator specifying which events to be audited (as specified in the Commercial Grade OS Requirement Set “8.1.2.2”) as the change of system database management -- audit trail setup and analysis. Therefore, the “8.1.2.4” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.1.2.2” requirement “The OS shall allow an authorized administrator to specify which events, from the set of auditable events, are to be audited”.

The Windows OS also addresses the “8.1.2.4” requirement because of the generation of the following two audit records:

* “[Event ID 4719](http://support.microsoft.com/kb/947226) “System audit policy was changed”;
* “[Event ID 4712](http://support.microsoft.com/kb/947226) “Per User Audit Policy was changed”.

Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.1.2.2” requirement for detail.

## Addressing 8.1.2.5 “The OS shall provide a protected mechanism for opening and closing audit”

The “8.1.2.5” requirement additionally requires that “the mechanism shall have the functions of selecting and changing audit events and set to default status when the system is running; the use of the mechanism shall be subject to restrictions of authorization by the system administrator; the system administrator shall be capable of selecting one or more authentication-based or object attribute-based user audit activities”

The “8.1.2.5” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.1.2.2” requirement “The OS shall allow an authorized administrator to specify which events, from the set of auditable events, are to be audited”.

The Windows OS also addresses the “8.1.2.5” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.1.2.2” requirement for detail.

## Addressing 8.1.2.6 “The protection to function related to audit trail and management shall be provided in order to fulfill the creation, destruction, emptying and archiving of audit trail”

Except the “creation” aspect, the “8.1.2.6” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.2.2.3” requirement “The OS shall provide an authorized administrator with the capability to archive audit data”.

Except the “creation” aspect, the Windows OS also addresses the “8.1.2.6” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.2.2.3” requirement for detail.

For the “creation” aspect, the “8.1.2.6” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.1.2.2” requirement “The OS shall allow an authorized administrator to specify which events, from the set of auditable events, are to be audited”.

For the “creation” aspect, the Windows OS also addresses the “8.1.2.6” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.1.2.2” requirement for detail.

## Addressing 8.1.2.7 “A mechanism shall be provided to show the current [audit policy] selection, and users of the [show the current [audit policy] selection] mechanism should be limited [to the] authorized users”

The Windows OS addresses the “8.1.2.7” requirement because of the availability of the following.

* The [LsarQueryAuditPolicy()](http://msdn.microsoft.com/en-us/library/aa375697(VS.85).aspx) interface from the Windows OS audit policy unit;
* The [system access control list (SACL)](http://msdn.microsoft.com/en-us/library/aa375723(VS.85).aspx) of [SYSTEM\_AUDIT\_ACE](http://msdn.microsoft.com/en-us/library/aa379616(VS.85).aspx) ACEs and [SYSTEM\_AUDIT\_OBJECT\_ACE](http://msdn.microsoft.com/en-us/library/aa379619(VS.85).aspx) ACEs within a [security descriptor structure](http://msdn2.microsoft.com/en-us/library/aa379561(VS.85).aspx).

To use the [LsarQueryAuditPolicy()](http://msdn.microsoft.com/en-us/library/aa375697(VS.85).aspx) interface to view an element of the audit policy, the caller subject must possess either the [SeSecurityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) or the [AUDIT\_QUERY\_SYSTEM\_POLICY](http://msdn.microsoft.com/en-us/library/aa375702(VS.85).aspx) permission right or the [AUDIT\_QUERY\_USER\_POLICY](http://msdn.microsoft.com/en-us/library/aa375697(VS.85).aspx) permission right. By default, the [SeSecurityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK) is enabled for any member of the “Administrators” local group. However, neither the [AUDIT\_QUERY\_SYSTEM\_POLICY](http://msdn.microsoft.com/en-us/library/aa375702(VS.85).aspx) permission right nor the [AUDIT\_QUERY\_USER\_POLICY](http://msdn.microsoft.com/en-us/library/aa375697(VS.85).aspx) permission right is assigned to any user account by default.

Viewing the [system access control list (SACL)](http://msdn.microsoft.com/en-us/library/aa375723(VS.85).aspx) of [SYSTEM\_AUDIT\_ACE](http://msdn.microsoft.com/en-us/library/aa379616(VS.85).aspx) ACEs and [SYSTEM\_AUDIT\_OBJECT\_ACE](http://msdn.microsoft.com/en-us/library/aa379619(VS.85).aspx) ACEs associated with a named object is possible within the Windows OS. This can be achieved by an authorized subject possessing the [SeSecurityPrivilege](http://msdn.microsoft.com/en-us/library/bb530716(VS.85).aspx?info=EXLINK).

# Meeting Additional “Auditing Audit Collection Audit Requirements”

The Commercial Grade OS Requirement Set already has 1 individual audit requirement under the heading of “Auditing Audit Collection Management Requirements”. It is listed as “8.1.3.1”.

There is 1 additional individual functional requirement under the heading of “Auditing Audit Collection Audit Requirements” in this appendix. It is listed as “8.1.3.2”.

## Addressing 8.1.3.2 “The OS shall point out the least types of recordable audit events, including the operations of system administrator and system security administrator and others”

The “8.1.3.2” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.1.3.1” requirement “The OS shall provide the ability to audit the modification of the threshold values specified in the Commercial Grade OS Requirement Set “8.1.1.3””.

The Windows OS also addresses the “8.1.3.2” requirement with the capabilities provided by [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx). Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.1.3.1” requirement for detail.

# Meeting Additional “Auditing Audit Storage Functional Requirements”

The Commercial Grade OS Requirement Set already has 2 individual functional requirements under the heading of “Auditing Audit Storage Functional Requirements”. They are listed as “8.2.1.n”, where n = 1, and 2.

There are 5 additional individual functional requirements under the heading of “Auditing Audit Storage Functional Requirements” in this appendix. They are listed as “8.2.1.n”, where n = 3, 4, 5, 6, and 7.

## Addressing 8.2.1.3 “The OS shall provide protected audit trail record”

The “8.2.1.3” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.2.1.1” requirement “The OS shall prevent modification of previously written audit records”.

The Windows OS also addresses the “8.2.1.3” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.2.1.1” requirement for detail.

## Addressing 8.2.1.4 “The audit process shall be generated, maintained and protected to avoid modification, unauthorized access and damage”

The “8.2.1.4” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.2.1.1” requirement “The OS shall prevent modification of previously written audit records”.

The Windows OS also addresses the “8.2.1.4” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.2.1.1” requirement for detail.

## Addressing 8.2.1.5 “The OS shall provide protection on audit records from being modified and destroyed by unauthorized users”

The “8.2.1.5” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.2.1.1” requirement “The OS shall prevent modification of previously written audit records”.

The Windows OS also addresses the “8.2.1.5” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.2.1.1” requirement for detail.

## Addressing 8.2.1.6 “When the audit is activated, the integrity of audit trail events shall be guaranteed (i.e. maintained)”

The “8.2.1.6” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.2.1.1” requirement “The OS shall prevent modification of previously written audit records”.

The Windows OS also addresses the “8.2.1.6” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.2.1.1” requirement for detail.

## Addressing 8.2.1.7 “Audit data which requires special protection shall be protected”

The “8.2.1.7” requirement is trivially satisfied due to the “8.2.1.4” requirement. As the “8.2.1.7” requirement does not clarify what audit data requires special protection and the “8.2.1.4” requirement already specifies the need of certain protection for all audit data, there is no need to provide additional protection which is different from the protection already required in the “8.2.1.4” requirement.

## Addressing 8.2.1.8 “Administrators should be able to define the thresholds beyond the limit of audit trail”

The “8.2.1.7” requirement additionally requires that “when storage space of the system is full, measure should be taken in accordance with the commands of system administrators, including alarm and discard unrecorded audit information, suspend audit, overwrite previous audit records, and others”.

Except the “in accordance with the commands of system administrators” aspect, the “8.2.1.7” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.2.1.2” requirement “The OS shall provide the capability for authorized administrators to specify the specific actions to be taken upon audit storage exhaustion”. The Commercial Grade OS Requirement Set suggests the following set of actions for an administrator to select:

1. stop performing operations that are being audited;
2. overwrite oldest audit data;
3. automatically increase audit storage space;
4. automatically archive audit data;
5. disable auditing and continue to operate.

Except the “in accordance with the commands of system administrators” aspect, the Windows OS also addresses the “8.2.1.7” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.2.1.2” requirement for detail.

For the “in accordance with the commands of system administrators” aspect, the “8.2.1.7” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.2.2.1” requirement “The OS shall provide an authorized administrator with the capability to specify actions to be taken upon audit storage exhaustion”.

For the “in accordance with the commands of system administrators” aspect, the Windows OS also addresses the “8.2.1.7” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.2.2.1” requirement for detail.

# Meeting Additional “Auditing Audit Storage Management Requirements”

The Commercial Grade OS Requirement Set already has 2 individual management requirements under the heading of “Auditing Audit Storage Management Requirements”. They are listed as “8.2.2.n”, where n = 1, 2, and 3.

There are 4 additional individual functional requirements under the heading of “Auditing Audit Storage Management Requirements” in this appendix. They are listed as “8.2.2.n”, where n = 4, 5, 6, and 7.

## Addressing 8.2.2.4 “The OS shall provide basic audit review, and limited audit review, optional audit inquiry”

The “8.2.2.4” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.2.2.2” requirement “The OS shall provide an authorized administrator with the capability to sort, select and review collected audit records based on identity and any audit information items specified in the Commercial Grade OS Requirement Set “8.1.1.4””.

The Windows OS also addresses the “8.2.2.4” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.2.2.2” requirement for detail.

## Addressing 8.2.2.5 “The access of unauthorized users to audit data shall be strictly restricted”

The “8.2.2.5” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.2.2.2” requirement “The OS shall provide an authorized administrator with the capability to sort, select and review collected audit records based on identity and any audit information items specified in the Commercial Grade OS Requirement Set “8.1.1.4””.

The Windows OS also addresses the “8.2.2.5” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.2.2.2” requirement for detail.

## Addressing 8.2.2.6 “Protection on audit records from being accessed by unauthorized users shall be provided”

The “8.2.2.6” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.2.2.2” requirement “The OS shall provide an authorized administrator with the capability to sort, select and review collected audit records based on identity and any audit information items specified in the Commercial Grade OS Requirement Set “8.1.1.4””.

The Windows OS also addresses the “8.2.2.6” requirement. Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.2.2.2” requirement for detail.

## Addressing 8.2.2.7 “Audit tools should be able to be authorized to an individual for supervision and browsing audit data”

The “8.2.2.7” requirement treats similar security concern(s) as the Commercial Grade OS Requirement Set “8.2.2.2” requirement “The OS shall provide an authorized administrator with the capability to sort, select and review collected audit records based on identity and any audit information items specified in the Commercial Grade OS Requirement Set “8.1.1.4””.

The Windows OS also addresses the “8.2.2.7” requirement with the [“event viewer” application (eventvwr.exe)](http://technet.microsoft.com/en-us/library/cc766401.aspx) and [Microsoft Operations Manager 2007](http://technet.microsoft.com/en-us/library/bb310604.aspx). Please see the justification text for addressing the Commercial Grade OS Requirement Set “8.2.2.2” requirement for detail.

The End

1. This is used by the boot manager to stop other boot applications the opportunity to use the TPM unsealing operation. By changing it, a subsequent use of the TPM unsealing operation would fail. [↑](#footnote-ref-2)
2. The Windows OS DACL mechanism will be discussed in more details in the “Addressing 1.2.3.1 “The OS shall provide the ability to audit when modification or insertion of security-relevant data received from a remote part of the OS has been detected”” section of this paper. [↑](#footnote-ref-3)
3. These credential verification specific components are the (default) security providers loaded by the Windows OS authentication service. [↑](#footnote-ref-4)
4. The Windows OS Local Security Account Management is a Windows OS service that maintains the database of the user accounts which are defined by a local administrator in the local Windows OS running on the local machine. [↑](#footnote-ref-5)
5. A Windows OS domain controller is a distributed part of the Windows OS Active Directory which maintains the user accounts which are defined by an enterprise administrator or a domain account operator (DOMAIN\_ALIAS\_RID\_ACCOUNT\_OPS) as directory objects in the Windows OS-enabled distributed OS environment configured for a Windows OS forest representing one or more domain naming contexts. [↑](#footnote-ref-6)
6. The DOMAIN\_LOCKOUT\_ADMINS policy causes the built-in administrator account to be locked out from network logons. [↑](#footnote-ref-7)
7. Here the caller is referred to the user account of the subject who requests to create the new user account. [↑](#footnote-ref-8)
8. Here the caller is referred to the user account of the subject who requests to create the new user account. [↑](#footnote-ref-9)
9. Here the caller is referred to the user account of the subject who requests to create the new user account. [↑](#footnote-ref-10)
10. Here the caller is referred to the user account of the subject who requests to create the new user account. [↑](#footnote-ref-11)
11. Here the caller is referred to the user account of the subject who requests to create the new user account. [↑](#footnote-ref-12)
12. Here the caller is referred to the user account of the subject who requests to create the new user account. [↑](#footnote-ref-13)
13. Here the caller is referred to the user account of the subject who requests to modify attributes of the user account. [↑](#footnote-ref-14)
14. Please also see the transition from the “Locked” state to the “Inactivity timeout handler during locked” state for the same security assertion. [↑](#footnote-ref-15)
15. Please also see the transition from the “Locked” state to the “Session disconnected during locked” state for the same security assertion. [↑](#footnote-ref-16)
16. Please also see the transition from the “Locked” state to the “Hibernate during locked” state for the same security assertion. [↑](#footnote-ref-17)
17. Please also see the transition from the “Locked” state to the “Session disconnected during locked” state for the same security assertion. [↑](#footnote-ref-18)
18. Please also see the transition from the “Locked” state to the “Hibernate during locked” state for the same security assertion. [↑](#footnote-ref-19)
19. Please also see the transition from the “Locked” state to the “Request to unlock” state for the same security assertion. [↑](#footnote-ref-20)
20. Please also see the transition from the “Locked” state to the “Welcome” state for the same security assertions. [↑](#footnote-ref-21)
21. Please also see the transition from the “Locked” state to the “Session disconnected during locked” state for the same security assertion. [↑](#footnote-ref-22)
22. Please also see the transition from the “Locked” state to the “Hibernate during locked” state for the same security assertion. [↑](#footnote-ref-23)
23. Please also see the transition from the “Locked” state to the “Session disconnected during locked” state for the same security assertion. [↑](#footnote-ref-24)
24. Please also see the transition from the “Locked” state to the “Hibernate during locked” state for the same security assertion. [↑](#footnote-ref-25)
25. Please also see the transition from the “User logged on” state to the “Welcome” state for the same security assertion. [↑](#footnote-ref-26)
26. Please also see the transition from the “User logged on” state to the “Session disconnected” state for the same security assertion. [↑](#footnote-ref-27)
27. Please also see the transition from the “User logged on” state to the “Post logged on hibernate” state for the same security assertion. [↑](#footnote-ref-28)
28. The imperfectness is due to the lack of the authenticity of the client source IP address through the authentication protocols such as Kerberos, NTLM, SSL/TLS, and Digest. [↑](#footnote-ref-29)
29. In the “[client side caching for offline files](http://msdn.microsoft.com/en-us/library/cc296090(VS.85).aspx)” scenario, named files from a network server are copied to the local Windows OS machine. The copied files are available for local access when the network to communicate with the network server is interrupted. When the network is interrupted, it is not the files of the network server that are attempted for access, but their local copies. Hence, the fresh copy of the security descriptor associated with a file of the network server is irrelevant in the “[client side caching for offline files](http://msdn.microsoft.com/en-us/library/cc296090(VS.85).aspx)” scenario. [↑](#footnote-ref-30)
30. This LsarSetAuditPolicy() interface should appear in [the “lsarpc” interface IDL file](http://msdn.microsoft.com/en-us/library/cc207224.aspx) for Windows Vista and Windows Server 2008. [↑](#footnote-ref-31)
31. This monitoring period of one second can be configured using the “FlushTimer” value under the “HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\WMI\Autologger\EventLog-Security” registry key. [↑](#footnote-ref-32)
32. This security context also includes the case of the unique built-in administrator maintained by the Windows OS. [↑](#footnote-ref-33)
33. This is the case where the user account is neither the built-in administrator nor a member of the “Administrators” group, but the user account has been granted some administrative privileges. [↑](#footnote-ref-34)
34. This prompting for user credentials is not meant to support an interactive user’s initial logon. It is triggered when certain running application requires the attention of the already logged on interactive user to supply the user’s input of his/her credentials for completing an application specific task. [↑](#footnote-ref-35)
35. The command launching information database stores the launching information associated with specific Windows OS commands which are meant to be launched during the next interactive user logon time or when the system restarts (i.e. reboots) the next time. [↑](#footnote-ref-36)
36. The soft audit storage is the audit storage that does not cause the local machine to prevent subsequent auditable events, except those taken by an administrator, to occur when the audit storage is full (i.e. reaching its maximum size). [↑](#footnote-ref-37)
37. A [file reference](http://msdn.microsoft.com/en-us/library/bb470206(VS.85).aspx) is the MFT segment reference of the base file record. It is of type FILE\_REFERENCE, which is also [MFT\_SEGMENT\_REFERENCE](http://msdn.microsoft.com/en-us/library/bb470211(VS.85).aspx). File references can be queried using the [FileIdBothDirectoryInformation](http://msdn.microsoft.com/en-us/library/cc205748(PROT.10).aspx) information class. [↑](#footnote-ref-38)
38. The soft audit storage is the audit storage that does not cause the local machine to prevent subsequent auditable events, except those taken by an administrator, to occur when the audit storage is full (i.e. reaching its maximum size). [↑](#footnote-ref-39)
39. The soft audit storage is the audit storage that does not cause the local machine to prevent subsequent auditable events, except those taken by an administrator, to occur when the audit storage is full (i.e. reaching its maximum size). [↑](#footnote-ref-40)