Microsoft Windows

Common Criteria Evaluation

Microsoft Windows 7

Microsoft Windows Server 2008 R2

Security Target

Document Information	
Version Number	1.0
Updated On	March 23, 2011

This is a preliminary document and may be changed substantially prior to final commercial release of the software described herein.

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

Complying with all applicable copyright laws is the responsibility of the user. This work is licensed under the Creative Commons Attribution-NoDerivs-NonCommercial License (which allows redistribution of the work). To view a copy of this license, visit http://creativecommons.org/licenses/by-nd-nc/1.0/ or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

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.

The example companies, organizations, products, people and events depicted herein are fictitious. No association with any real company, organization, product, person or event is intended or should be inferred.

© 2011 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, Visual Basic, Visual Studio, Windows, the Windows logo, Windows NT, and Windows Server are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

TABLE OF CONTENTS

LIST OF TABLES

1 Security Target Introduction

This section presents the following information required for a Common Criteria (CC) evaluation:

- Identifies the Security Target (ST) and Target of Evaluation (TOE);
- Specifies the ST conventions and ST conformance claims; and,
- Describes the ST organization.

1.1 Security Target, TOE, and Common Criteria (CC) Identification

ST Title: Microsoft Windows 7 and Windows Server 2008 R2 Security Target

ST Version: Version 1.0, March 23, 2011

TOE Software Identification: The following Windows Operating Systems (OS):

- Microsoft Windows 7 Enterprise Edition (32-bit and 64-bit versions)
- Microsoft Windows 7 Ultimate Edition (32-bit and 64-bit versions)
- Microsoft Windows Server 2008 R2 Standard Edition
- Microsoft Windows Server 2008 R2 Enterprise Edition
- Microsoft Windows Server 2008 R2 Datacenter Edition
- Microsoft Windows Server 2008 R2 Itanium Edition

The following security updates and patches must be applied to the above Windows 7 products:

- All security updates as of September 14, 2010 as well as the updates associated with security bulletins MS10-073 and MS10-085
- Hotfix KB2492505

The following security updates must be applied to the above Windows Server 2008 R2 products:

- All security updates as of September 14, 2010 as well as the updates associated with security bulletins MS10-073 and MS10-085
- Hotfix KB2492505

TOE Hardware Identification – The following real and virtualized hardware platforms and components are included in the evaluated configuration:

- Dell Optiplex 755, 3.0 GHz Intel Core 2 Duo E8400, 64-bit
- Dell PowerEdge SC1420, 3.6 GHz Intel Xeon Processor (1 CPU), 64-bit
- Dell PowerEdge 2970, 1.7 GHz quad core AMD Opteron 2344 Processor (2 CPUs), 64-bit
- HP Proliant DL385 G5, 2.1 GHz quad core AMD Opteron 2352 Processor (2 CPUs), 64-bit
- HP Proliant DL385, 2.6 GHz AMD Opteron 252 Processor (2 CPUs), 64-bit
- HP Integrity rx1620, 1.3 GHz Intel Itanium Processor (1 CPU), 64-bit (Itanium)

- Microsoft Hyper-V
- Microelectronics Trusted Platform Module [SMO1200]
- GemPlus GemPC Twin USB smart card reader

TOE Guidance Identification – The following administrator, user, and configuration guides were evaluated as part of the TOE:

> *Microsoft Windows Common Criteria Evaluation, Microsoft Windows 7/Microsoft Windows Server 2008 R2, Common Criteria Supplemental Admin Guidance* along with all the documents referenced therein including the *Windows 7 Security and Windows Server 2008 R2 Security Guides* published by Microsoft. Among other things identified in the evaluated guides, the BitLocker security feature must be configured in the evaluated configuration.

Evaluation Assurance Level (EAL): EAL 4 augmented with ALC_FLR.3 (Systematic Flaw Remediation).

CC Identification: CC for Information Technology (IT) Security Evaluation, Version 3.1, Revision 3, July 2009.

1.2 CC Conformance Claims

This TOE and ST are consistent with the following specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 3.1, Revision 3, July 2009, extended (Part 2 extended)
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements Version 3.1, Revision 3, July 2009, conformant (Part 3 conformant)
- Evaluation Assurance Level 4 (EAL4) augmented with ALC_FLR.3 (EAL 4 augmented)
- US Government Protection Profile for General-Purpose Operating Systems in a Networked environment (GPOSPP), version 1.0, 30 August 2010 augmented (with EAL4 and ALC_FLR.3 and other security functional requirements identified later)

1.3 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the ST.

1.3.1 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements (SFRs): Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations.
    - Assignment: allows the specification of an identified parameter.
    - Selection: allows the specification of one or more elements from a list.
    - Refinement: allows the addition of details.

The conventions for the assignment, selection, refinement, and iteration operations are described in Section 5.

- Other sections of the ST – Other sections of the ST use a bold font to highlight text of special interest, such as captions.

1.3.2 Terminology

The following terminology is used in the security target (ST):

Term	Definition
Access	Interaction between an entity and an object that results in the flow or modification of data.
Access control	Security service that controls the use of resources[1] and the disclosure and modification of data[2].
Accountability	Tracing each activity in an IT system to the entity responsible for the activity.
Administrator	An authorized user who has been specifically granted the authority to manage some portion or the entire TOE and thus whose actions may affect the TSP. Administrators may possess special privileges that provide capabilities to override portions of the TOE Security Policy (TSP).
Assurance	A measure of confidence that the security features of an IT system are sufficient to enforce it's' security policy.
Attack	An intentional act attempting to violate the security policy of an IT system.
Authentication	Security measure that verifies a claimed identity.
Authentication data	Information used to verify a claimed identity.
Authorization	Permission, granted by an entity authorized to do so, to perform functions and access data.
Authorized user	An authenticated user who may, in accordance with the TSP, perform an operation.
Availability	Timely[3], reliable access to IT resources.
Compromise	Violation of a security policy.
Confidentiality	A security policy pertaining to disclosure of data.
Critical cryptographic security parameters	Security-related information appearing in plaintext or otherwise unprotected form and whose disclosure or modification can compromise the security of a cryptographic module or the security of the information protected by the module.
Cryptographic boundary	An explicitly defined contiguous perimeter that establishes the physical bounds (for hardware) or logical bounds (for software) of a cryptographic module.
Cryptographic key (key)	A parameter used in conjunction with a cryptographic algorithm that determines: • the transformation of plaintext data into ciphertext data,

[1] Hardware and software
[2] Stored or communicated
[3] According to a defined metric

Term	Definition
	• the transformation of ciphertext data into plaintext data, • a digital signature computed from data, • the verification of a digital signature computed from data, or • a data authentication code computed from data.
Cryptographic module	The set of hardware, software, and/or firmware that implements approved security functions (including cryptographic algorithms and key generation) and is contained within the cryptographic boundary.
Cryptographic module security policy	A precise specification of the security rules under which a cryptographic module must operate.
Defense-in-depth	A security design strategy whereby layers of protection are utilized to establish an adequate security posture for an IT system.
Discretionary Access Control (DAC)	A means of restricting access to objects based on the identity of subjects and groups to which they belong. The controls are discretionary in the sense that a subject with a certain access permission is capable of passing that permission (perhaps indirectly) on to any other subject.
Enclave	A collection of entities under the control of a single authority and having a homogeneous security policy. They may be logical, or based on physical location and proximity.
Entity	A subject, object, user or external IT device.
General-Purpose Operating System	A general-purpose operating system is designed to meet a variety of goals, including protection between users and applications, fast response time for interactive applications, high throughput for server applications, and high overall resource utilization.
Identity	A means of uniquely identifying an authorized user of the TOE.
Named object	An object that exhibits all of the following characteristics: • The object may be used to transfer information between subjects of differing user identities within the TOE Security Function (TSF). • Subjects in the TOE must be able to request a specific instance of the object. • The name used to refer to a specific instance of the object must exist in a context that potentially allows subjects with different user identities to request the same instance of the object.
NAP-NAC	Network Access Protection (NAP) used in conjunction with Cisco Network Admission Control (NAC) to offer the ability to restrict access to network resources.
NAP Agent	A service included with Windows 7, Windows Server 2008 and R2, Windows Vista, and Windows XP with SP3 that collects and manages health information for NAP client computers.
NAP client computer	A computer that has the NAP Agent service installed and running, and provides its health status to NAP server computers.
NAP-capable computer	A computer that has the NAP Agent service installed and running and is capable of providing its health status to NAP server computers. NAP-capable computers include computers running Windows 7, Windows Server 2008 and R2, Windows Vista, and Windows XP with SP3.
(NAP) Non-NAP-capable	A computer that cannot provide its health status to NAP server

Term	Definition
computer	components. A computer that has NAP agent installed but not running is also considered non-NAP-capable.
(NAP) Compliant computer	A computer that meets the NAP health requirements that you have defined for the network. Only NAP client computers can be compliant.
(NAP) Noncompliant computer	A computer that does not meet the NAP health requirements defined for the network. Only NAP client computers can be noncompliant.
(NAP) Health status	Information about a NAP client computer that NAP uses to allow or restrict access to a network. Health is defined by a client computer's configuration state. Some common measurements of health include the operational status of the Windows Firewall, the update status of antivirus signatures, and the installation status of security updates. A NAP client computer provides health status by sending a message called a statement of health (SoH).
NAP health policy server	A NAP health policy server is a computer running Windows Server 2008 and R2 with the Network Policy Server (NPS) role service installed and configured for NAP. The NAP health policy server uses NPS policies and settings to evaluate the health of NAP client computers when they request access to the network, or when their health state changes. Based on the results of this evaluation, the NAP health policy server instructs whether NAP client computers will be granted full or restricted access to the network.
Object	An entity under the control of the TOE that contains or receives information and upon which subjects perform operations.
Operating environment	The total environment in which a TOE operates. It includes the physical facility and any physical, procedural, administrative and personnel controls.
Persistent storage	All types of data storage media that maintain data across system boots (e.g., hard disk, removable media).
Public object	An object for which the TSF unconditionally permits all entities “read” access. Only the TSF or authorized administrators may create, delete, or modify the public objects.
Resource	A fundamental element in an IT system (e.g., processing time, disk space, and memory) that may be used to create the abstractions of subjects and objects.
Secure State	Condition in which all TOE security policies are enforced.
Security attributes	TSF data associated with subjects, objects and users that is used for the enforcement of the TSP.
Security-enforcing	A term used to indicate that the entity (e.g., module, interface, subsystem) is related to the enforcement of the TOE security policies.
Security-supporting	A term used to indicate that the entity (e.g., module, interface, subsystem) is not security-enforcing however, its implementation must still preserve the security of the TSF.
Security Target (ST)	A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.
Subject	An active entity within the TOE Scope of Control (TSC) that causes

Term	Definition
	operations to be performed. Subjects can come in two forms: trusted and untrusted. Trusted subjects are exempt from part or all of the TOE security policies. Untrusted subjects are bound by all TOE security policies.
Target of Evaluation (TOE)	An IT product or system and its associated administrator and user guidance documentation that is the subject of an evaluation.
Threat	Capabilities, intentions and attack methods of adversaries, or any circumstance or event, with the potential to violate the TOE security policy.
User	Any person who interacts with the TOE.
Vulnerability	A weakness that can be exploited to violate the TOE security policy.

1.3.3 Acronyms

The acronyms used in this security target are specified in **Appendix A — List of Abbreviations**.

1.4 ST Overview and Organization

The Windows 7 and Windows Server 2008 R2 TOE is a general-purpose, distributed, network OS that provides controlled access between subjects and user data objects. The Windows 7 and Windows Server 2008 R2 TOE has a broad set of security capabilities including

- single network logon (using password or smart card)
- access control and data encryption
- extensive security audit collection
- host-based firewall and IPSec to control information flow
- public key certificate service
- built-in standard-based security protocols such as
  - Kerberos[4]
  - Transport Layer Security (TLS)/Secure Sockets Layer (SSL)[5]
  - Digest[6]
  - Internet Key Exchange (IKE)/IPSec[7]
- Light-weight Directory Access Protocol (LDAP) Directory-based resource management[8]
- FIPS-140 validated cryptography

[4] See http://msdn.microsoft.com/en-us/library/cc233855(PROT.10).aspx for more information about the Windows implementation of Kerberos.

[5] See http://msdn.microsoft.com/en-us/library/dd207968(PROT.10).aspx for more information about the Windows implementation of TLS/SSL.

[6] See http://msdn.microsoft.com/en-us/library/cc227906(PROT.10).aspx for more information about the Windows implementation of Digest authentication.

[7] See http://msdn.microsoft.com/en-us/library/cc233219(PROT.10).aspx for more information about the Windows implementation of IKE and IPSec.

[8] See http://msdn.microsoft.com/en-us/library/cc223122(PROT.10).aspx for more information about the Windows implementation of LDAP.

The Windows 7 and Windows Server 2008 R2 TOE provides the following security services

- User data protection (DAC, IPSec information flow control, connection firewall information flow control, WEBUSER access control, web content provider access control,)
- Cryptographic support
- Audit
- Identification and Authentication (I&A) (including trusted path/channel)
- Security management
- Protection of the TOE Security Functions (TSF)
- Resource quotas
- TOE access/session control

The Windows 7 and Windows Server 2008 R2 TOE security policies provide network-wide controlled access protection (access control for user data, WEBUSER and web content provider, IPSec information flow, connection firewall information flow), encrypted data/key protection, and encrypted file protection.

These policies enforce access limitations between individual users and data objects, and on in-coming and out-going traffic channels through physically separated parts of the TOE. The TOE is capable of auditing security relevant events that occur within a Windows 7 and Windows Server 2008 R2 network. All these security controls require users to identify themselves and be authenticated prior to using any node on the network.

The Windows 7 and Windows Server 2008 R2 ST contains the following additional sections:

- TOE Description (Section 2): Provides an overview of the TSF and boundary.
- Security Problem Definition (Section 3) : Describes the threats, organizational security policies and assumptions that pertain to the TOE.
- Security Objectives (Section 4): Identifies the security objectives that are satisfied by the TOE and the TOE operational environment.
- Security Requirements (Section 5): Presents the security functional and assurance requirements met by the TOE.
- TOE Summary Specification (Section 6): Describes the security functions provided by the TOE to satisfy the security requirements and objectives.
- PP Claims (Section 7): Presents the rationale concerning compliance of the ST with the *US Government Protection Profile for General-Purpose Operating Systems in a Networked Environment.*
- Rationale (Section 8): Presents the rationale for the security objectives, requirements, and TOE Summary Specifications (TSS) as to their consistency, completeness and suitability.

2 TOE Description

The TOE includes the Windows 7 operating system, the Microsoft Windows Server® 2008 R2 operating system, supporting hardware, and those applications necessary to manage, support and configure the OS.

2.1 Product Types

Windows 7 and Windows Server 2008 R2 are a preemptive multitasking, multiprocessor, and multi-user operating systems. In general, operating systems provide users with a convenient interface to manage underlying hardware. They control the allocation and manage computing resources such as processors, memory, and Input/Output (I/O) devices. Windows 7 and Windows Server 2008 R2 expand these basic operating system capabilities to controlling the allocation and managing higher level IT resources such as security principals (user or machine accounts), files, printing objects, services, windowstation, desktops, cryptographic keys, network ports/traffics, directory objects, and web content. Multi-user operating systems such as Windows 7 and Windows Server 2008 R2, keep track of which user is using which resource, grant resource requests, account for resource usage, and mediate conflicting requests from different programs and users.

Windows 7 and Windows Server 2008 R2 provide an interactive User Interface (UI), as well as a network interface. The TOE includes a set of Windows 7 and Windows Server 2008 R2 systems that can be connected via their network interfaces and may be organized into domains and forests. A domain is a logical collection of Windows 7 and Windows Server 2008 R2 systems that allows the administration and application of a common security policy and the use of a common accounts database. One or more domains combines to comprise a forest. Windows 7 and Windows Server 2008 R2 support single-domain and multiple-domain (i.e., forest) configurations.

Each domain must include at least one designated server known as a Domain Controller (DC) to manage the domain. The TOE allows for multiple DCs that replicate TOE data among themselves to provide for higher availability.

Each Windows 7 and Windows Server 2008 R2 system, whether it is a DC server, non-DC server, or workstation, is part of the TOE and provides a subset of the TSFs. The TSF for Windows 7 and Windows Server 2008 R2 can consist of the security functions from a single system (in the case of a stand-alone system) or the collection of security functions from an entire network of systems (in the case of domain configurations).

Within this Security Target, when specifically referring to a type of TSF (e.g., DC), the TSF type will be explicitly stated. Otherwise, the term TSF refers to the total of all TSFs within the TOE.

In addition to core **operating system** capabilities Windows 7 and Windows Server 2008 R2 can also be categorized as the following types of Information Assurance (IA) or IA-enabled IT products:

- Windows 7 and Windows Server 2008 R2 serve as a **Sensitive Data Protection Device** to defend the computing environment. The core mechanisms in this case are BitLocker and the Encrypting File System (EFS), which are part of the Windows 7 and Windows Server 2008 R2 TOE.

- Windows Server 2008 R2 is a **Directory Service** product to support security infrastructure. The LDAP-based access and management of Windows Active Directory (AD) objects are part of the Windows Server 2008 R2 TSF Interfaces (TSFI). Note that Windows 7 includes the capability to act as a Directory Service client.
- Windows 7 and Windows Server 2008 R2 is a **Network Management** product to support security infrastructure. Group Policy, which is part of the Windows 7 and Windows Server 2008 R2 TOE, provide the network management in Windows 7 and Windows Server 2008 R2 networks.
- Windows 7 and Windows Server 2008 R2 is a **Desktop Management** product to support security infrastructure. The Group Policy Service, which is part of Windows 7 and Windows Server 2008 R2 TOE, provides the desktop management of Windows 7 and Windows Server 2008 R2 TOE desktops.
- Windows 7 and Windows Server 2008 R2 is a **Single Sign-On** product (using password or smart card) for Windows 7 and Windows Server 2008 R2 networks to defend the computing environment. Windows 7 and Windows Server 2008 R2 support single sign on to the TOE.
- Windows 7 and Windows Server 2008 R2 is a **Firewall** (Network and Host-based) product with the capability to filter network traffic based upon source and destination addresses/ports and protocol.
- Windows 7 and Windows Server 2008 R2 is a **VPN** product providing an IPSec service and its associated Transport Driver Interface (TDI) based network support.
- Windows Server 2008 is a **Web Server** product by including the Internet Information Services (IIS) component functionality which provides a web service application infrastructure utilizing the underlying OS services.

2.2 Product Description

Windows 7 and Windows Server 2008 R2 are operating systems that support both workstation and server installations. The TOE includes six product variants of Windows 7 and Windows Server 2008 R2:

- Windows 7 Enterprise
- Windows 7 Ultimate
- Windows Server 2008 R2 Standard
- Windows Server 2008 R2 Enterprise
- Windows Server 2008 R2 Datacenter
- Windows Server 2008 R2 Itanium

Windows 7 is suited for business desktops and notebook computers. It is the workstation product and while it can be used by itself, it is designed to serve as a client within Windows domains.

Built for departmental and standard workloads, Windows Server 2008 R2 Standard delivers intelligent file and printer sharing; secure connectivity based on Internet technologies, and centralized desktop policy management.

Windows Server 2008 R2 Enterprise differs from Windows Server 2008 R2 Standard primarily in its support for high-performance server hardware for greater load handling and additional server roles. These capabilities provide reliability that helps ensure systems remain available.

Windows Server 2008 R2 Datacenter provides the necessary scalable and reliable foundation to support mission-critical solutions for databases, enterprise resource planning software, high-volume, real-time transaction processing, and server consolidation.

Windows Server 2008 R2 Itanium provides support for the alternate Intel Itanium CPU, but otherwise can serve where Standard or Enterprise edition products might be used.

In terms of security, Windows 7 and Server 2008 R2 share the same security characteristics. The primary difference is that the Server 2008 Server R2 products include services and capabilities that are not part of Windows 7 (for example the DNS Server, DHCP Server) or are not installed by default on Server 2008 R2 (for example the Windows Media Player, Windows Aero and desktop themes). The additional services have a bearing on the security properties of the distributed operating system (e.g., by extending the set of available interfaces and proffered services) and as such are included within the scope of the evaluation.

2.2.1 Services and Capabilities in Windows Server 2008 R2

This section describes some of the additional services and capabilities in the server operating systems.

2.2.1.1 Hardware Capabilities

One differentiator between Windows Server editions is support for additional scalability and hardware capabilities. The following table states which hardware capabilities are supported by each edition of Windows Server 2008 R2.

Table 2-1 Hardware Capabilities for Windows Server 2008 R2

	Windows Server 2008 R2 Edition			
Capability	Standard	Enterprise	Datacenter	Itanium
Maximum Memory (RAM)	32 GB	2 TB	2 TB	2 TB
Maximum # of Processors	4 x 64	8 x64	64 x64	64 IA 64
Clustering	No	16-node	16-node	8-node
Hot Add/Replace Memory and Processors[9]	No	Yes	Yes	Yes
Fault-tolerant Memory Synchronization	No	Yes	Yes	Yes

2.2.1.2 Software Capabilities

Starting with Windows Server 2008, the server operating system was split into multiple server roles, with each server role providing different services and capabilities. This componentization simplifies administration and also reduces the attack surface of Windows Server by enabling the administrator to install only the specific binaries needed onto a machine to fulfill its role.

[9] Requires supporting hardware.

The following table indicates which roles are included in each edition of Windows Server:

Table 2-2 Server Roles in Windows Server 2008 R2

Server Role	Windows Server 2008 R2 Edition			
	Standard	Enterprise	Datacenter	Itanium
Active Directory Certificate Services	Yes[10]	Yes	Yes	
Active Directory Domain Services	Yes	Yes	Yes	
Active Directory Federation Services		Yes	Yes	
Active Directory Lightweight Directory Services	Yes	Yes	Yes	
Active Directory Rights Management Services	Yes	Yes	Yes	
Application Server	Yes	Yes	Yes	Yes
DHCP Server	Yes	Yes	Yes	
DNS Server	Yes	Yes	Yes	
Fax Server	Yes	Yes	Yes	
File Services	Yes[11]	Yes	Yes	
Hyper-V[12]	Yes	Yes	Yes	
Network Policy and Access Services	Yes[13]	Yes	Yes	
Print and Document Services	Yes	Yes	Yes	
Remote Desktop Services	Yes[14]	Yes	Yes	
Web Services (IIS 7.5)	Yes	Yes	Yes	Yes
Windows Deployment Services	Yes	Yes	Yes	
Windows Server Update Services (WSUS)	Yes	Yes	Yes	

Additionally all editions of Windows server include the Server Manager application which administrators use to add/remove roles and features from Windows Server as well as the Server Core, which a minimal server installation option for computers running on the Windows Server 2008 R2 operating system. Server Core provides a low-maintenance server environment with reduced attack surface by presenting a command-line local interface to the administrator instead of the GUI-based Explorer interface.

The security features addressed by this security target are those provided by Windows 7 and Windows Server 2008 R2 as operating systems. Microsoft provides several Window 7 and Windows Server 2008 R2 software applications that are considered outside the scope of the defined TOE and thus not part of the evaluated configuration. Services outside this evaluation include: e-mail service (SMTP), Remote Desktop, Rights Management Service, Windows SharePoint Service, Microsoft Message Queuing, and ReadyBoost. These services are particularly complex and in some cases essentially represent products in

[10] Limited to creating non-Enterprise Certificate Authorities. Also, does not support role separation.
[11] Limited to 1 standalone DFS root.
[12] Server 2008 Hyper-V was part of a separate Common Criteria evaluation.
[13] Limited to 250 Routing and Remote Access (RRAS) connections, 50 (Internet Authentication Service) IAS connections and 2 IAS Server Groups.
[14] Limited to 250 Remote Desktop Services connections.

their own right. They have been excluded because they are not enabled or installed by default and are not necessary for the operation of the core security services. Also they may have significant impact on the claims made in this Security Target and the ability of the TOE to conform to the intended Protection Profile.

While the Windows CC evaluation includes the IIS web server, the evaluated configuration does not allow for arbitrary server-side execution of web content (per the configuration guidance) since user subject binding would be uncertain. Similarly, the Network Access Protection (NAP) features related to 802.1X and NAP-NAC (see below) are excluded from the evaluated configuration since wireless technology and Cisco products are not included in the scope of the Microsoft Windows CC evaluation.

The following table summarizes the Windows configurations included in the evaluation.

	Windows 7 Enterprise	Windows 7 Ultimate	Windows Server 2008 R2 Standard	Windows Server 2008 R2 Enterprise	Windows Server 2008 R2 Datacenter	Windows Server 2008 R2 Itanium
32-bit	X	X	N/A	N/A	N/A	N/A
64-bit	X	X	X	X	X	X
Single Core/Processor	X	X	X	X	X	X
Multiple Core/Processor	X	X	X	X	X	X
Domain Member	X	X	X	X	X	X
Domain Controller	N/A	N/A	X	X	X	N/A

Table 2-3 Evaluated Configurations of Windows

2.3 Product Features

Windows 7 and Windows Server 2008 R2 have many features, several of which support simplifying the administration and management of a distributed computing environment, that improve network security and scalability. This section highlights several of these features while distinguishing those new to this evaluation as opposed to those features, albeit perhaps changed, subject to a previous evaluation.

2.3.1 New Security Features

The following features were not available in previous Windows operating system CC evaluations, but are included in this evaluation of Windows 7 and Windows Server 2008 R2.

BitLocker to Go

Windows 7 and Windows Server 2008 R2 extend the previous BitLocker data protection capabilities with the ability to also encrypt removable USB storage devices (e.g., USB flash drives). The removable USB storage device content can be encrypted using either a password or credentials on a smart card.

When a password is used, a version of the BitLocker To Go (BTG) Reader application (that is capable of providing read access to the encrypted content when the appropriate credentials are provided) is placed onto removable USB storage devices that are configured to use this feature. While the content of a removable USB storage device can be read and written when using Windows 7 or Windows Server 2008 R2 (assuming appropriate credentials are available), the BitLocker Reader application provides a read-only dialog that allows content to be copied via the application, when provided the correct password, to the host operating system so that the decrypted file content can be accessed.

Note that while the ability to encrypt content placed on the USB device is within scope of the evaluation, the BitLocker Reader is not considered part of the TOE Security Functions since it cannot be reliably protected and as such could potentially be modified or replaced (by the user or anyone else that may come into possession of the USB device).

Additionally, the Group Policy can be used to configure USB storage devices to effectively require BitLocker To Go to be used in order to write content on removable USB storage devices. Otherwise, such devices can be only used for read-only access.

DirectAccess

Windows 7 and Windows Server 2008 R2 introduce DirectAccess. DirectAccess allows clients to securely access file shares, web sites, and applications without connection to a virtual private network (VPN). DirectAccess involves the establishment of bi-directional communication paths between applicable Windows operating systems when suitable network connectivity (e.g., to the Internet) exists.

This feature is based on other features identified below, primarily IPSec and IPv6.

Network Access Protection

While present in previous versions of Windows, the Network Access Protection (NAP) feature has not previously been subject to Common Criteria evaluation. This feature allows access to network resources to be controlled based on a computer’s identity and compliance with configurable governance policies. The NAP mechanism is capable of automatically bringing a client workstation or server into compliance with defined governance policies so that access is subsequently allowed.

The NAP feature involves a NAP agent running on NAP clients and a NAP health policy server (NAP server) running on a Windows 2008 R2 server, with the Network Policy Server (NPS) role. The NAP agents collect relevant health information for their host NAP client and provide it to the NAP server when network access is required. The NAP server uses NPS policies and settings to evaluate the health of NAP clients in order to determine whether to grant network access (full or restricted). When a NAP client is not conformant with configured settings, and policies only restricted network access are allowed. However, the NAP server and NAP agent can cooperate to remedy some identified problems in order to bring a NAP client into compliance so that its network access can be elevated.

Access to a network subsequent to NAP server approval can be enforced using the following mechanisms: IPsec, VPN, DHCP, 802.1X and NAP-NAC (this last mechanism applies only when suitable Cisco devices are employed).[15]

2.3.2 Previously Evaluated Security Features

Windows 7 and Windows Server 2008 R2 provide a wide range of security features including flexible security management features, data and network protection features, and scalability features among others. Note that while some of the following features may not be obviously or directly security relevant, they do serve to preserve the security represented in the claims of this Security Target.

Access Control Lists

Windows 7 and Windows Server 2008 R2 permit only authenticated users to access system resources. The security model includes components to control who accesses objects (such as files, directories, and shared printers), what actions an individual can perform with respect to an object, and the events that are audited.

Every object has a unique Security Descriptor (SD) that includes an Access Control List (ACL). An ACL is a list of entries that grant or deny specific access rights to individuals or groups. The Windows 7 and Windows Server 2008 R2 object-based security model lets administrators grant access rights to a user or group that govern who can access a specific object managed by the local computer.

In distributed Windows deployments, administrators use the same object-based security model to grant access rights to users and groups managed that are managed by the Active Directory for (1) Active Directory objects, (2) sets of properties on an Active Directory object, and (3) individual properties of an Active Directory object. The definition of access rights on a per-property level provides the highest level of granularity of permissions for the Active Directory object.[16]

Auto-enrollment

[15] Enforcing access control for 802.1X and NAP-NAC were not included in the evaluated configuration because those mechanisms rely on non-Microsoft devices that are outside the scope of the evaluation.

[16] The term “property” for an Active Directory object is analogous to an “attribute” for an object in a directory service like X.500. In Windows operating systems, Active Directory objects are the only type of named object that contain properties which also have an associated ACL.

Public Key Certificate auto-enrollment and auto-renewal in Windows Server 2008 R2 significantly reduce the resources needed to manage x.509 certificates. These features also make it easier to deploy smart cards faster, and to improve the security of the Windows PKI by automatically expiring and renewing certificates.

Background Intelligent Transfer Service (BITS)

Windows 7 and Windows Server 2008 R2 expose a feature via Component Object Model (COM) to transfer data in a prioritized, throttled, and asynchronous manner between connected systems using idle network bandwidth. The Background Intelligent Transfer Service (BITS) protocol downloads content via HTTP and relies on HTTPS for data integrity. Windows uses BITS to download security updates for Windows from an update server.

Client Side Caching of Off-line Files for SMB/Common Internet File System (CIFS)

When Windows 7 and Windows Server 2008 R2 client caches a file and the file server is available, the client with the SMB/CIFS Redirector checks with the file server to verify that the cached version of the file is up-to-date. If the file is up-to-date, then the client uses the cached copy of the file. Note that this check involves not only the content of the file, but also all of the file's attributes (e.g., its security descriptor). If the Windows 7 and Windows Server 2008 R2 file server is not available, the client with the SMB/CIFS Redirector also has the cached copy to use.

Code Integrity Verification

Kernel-mode code signing (KMCS) prevents kernel-mode device drivers from loading unless they are published and digitally signed by developers who have been vetted by one of a handful of trusted certificate authorities (CAs). KMCS, using public-key cryptography technologies, requires that kernel-mode code include a digital signature generated by one of the trusted certificate authorities. When a Windows device driver tries to load, the TOE decrypts the hash included with the code using the public key stored in the certificate, then verifies that the hash matches the one computed with the code. The authenticity of the certificate is checked in the same way, but using the certificate authority's public key, which is trusted by Windows.

COM Plus Component Service Infrastructure

COM Plus Component Service is an Infrastructure running the Windows 7 and Windows Server 2008 R2 TOE based on extensions of the Component Object Model (COM). COM Plus Component Service provides the COM runtime environment with threading and security, object pooling, queued components, and application administration and packaging.

Delegation

Delegation is the act of allowing a Windows service to impersonate a user account or computer account in order to access resources throughout the network. This feature in Windows Server 2008 R2 enables you to limit delegation to specific services, to control the particular network resources the service or computer can use. For example, a service that was previously trusted for delegation in order to access a backend server on behalf of a user can now be constrained to use its delegation privilege only to that backend server and not to other machines or services.

Credential Manager

This provides a secure store for usernames/passwords and also stores links to certificates and keys. This enables a consistent single sign-on experience for users, including roaming users. Single sign-on makes it possible for users to access resources over the network without having to repeatedly supply their credentials.

Cross–Certification Support

Also called qualified subordination[17], Cross-Certification allows constraints to be placed on subordinate Certificate Authorities (CAs) and on the certificates they issue, and allows trust to be established between CAs in separate hierarchies. Cross-Certification support improves the efficiency of administering PKI.

Cryptographic API: Next Generation

Windows 7 and Windows Server 2008 R2 supplement the legacy CryptoAPI with the Cryptography API: Next Generation (CNG). CNG provides applications with access to cryptographic functions, public keys, credential management and certificate validation functions, and provides support for the National Security Agency's Suite B cryptographic algorithms. CNG also provides extensive auditing support, support for replaceable random number generators; keys are managed within a key isolation service to limit the exposure of secret and private keys.

Data Protection

Windows 7 and Windows Server 2008 R2 have improved support for data protection at the file, directory, and machine level.

The Encrypting File System (EFS) provides user-based file and directory encryption and has been enhanced to allow storage of encryption keys on smart cards, providing better protection of encryption keys.

The BitLocker Drive Encryption enterprise feature adds machine-level data protection. On a computer with appropriate hardware (e.g., Trusted Platform Module (TPM) support), BitLocker Drive Encryption provides full volume encryption of the system volume, including Windows system files and the hibernation file, which helps protect data from being compromised on a lost or stolen machine.

BitLocker also stores measurements (hashes) of core operating system files used during the early stages of initialization. Every time the computer is started, Windows verifies that the operating system files have not been modified outside of Windows control. If the files have been modified, Windows alerts the

[17] Qualified subordination is different from "qualified certificates" defined in RFC 3739.

user and then goes into a recovery mode, prompting the user to provide a recovery key (created previously when BitLocker was configured) to allow access to the encrypted disk volume.

Delegated Administration

Windows includes Active Directory (AD), a scalable, standard-compliant directory service. AD centrally manages Windows-based clients and servers, through a single consistent management interface, reducing redundancy and maintenance costs.

AD enables authorized administrators to delegate a selected set of administrative privileges to appropriate individuals within the organization to distribute the management and improve accuracy of administration. Delegated Administration helps companies reduce the number of domains they need to support a large organization with multiple geographical locations by allowing the delegation of only appropriate authorities, as opposed to creating new domains in order to define and limit the scope of administrative authorities.

AD can interoperate or synchronize data with other directory services using LDAP.

Delta Certificate Revocation Lists (CRLs)

The certificate server included in Windows Server 2008 R2 TOE supports Delta CRL, which makes publication of revoked X.509 certificates more efficient. A Delta CRL is a list containing only certificates whose status has changed since the last full (base) CRL was compiled. This is a much smaller object than a full CRL and can be published frequently with little or no impact on client machines or network infrastructure.

Digest Authentication

Digest authentication operates much like Basic authentication. However, unlike Basic authentication, Digest authentication transmits credentials across the network as a hash value, also known as a message digest. The user name and password cannot be deciphered from the hash value. Conversely, Basic authentication sends a Base 64 encoded password, essentially in clear text, across the network. Basic authentication is not supported in the TOE. Digest authentication does not have to use reversible password encryption. The AD extended schema properties ensures that every newly created user account automatically has the Digest authentication password hashed and stored as a field in the “AltSecId” property of the user object. Note that the hash is protected from replay using a challenge response protocol to introduce some unpredictable data.

Disk Quotas

Windows 7 and Windows Server 2008 R2 allow authorized administrators to set quotas on disk space usage per user and per volume to provide increased availability of disk space and help capacity planning efforts.

Distributed File System

Windows 7 and Windows Server 2008 R2 DFS builds a single, hierarchical view of multiple file servers and file server shares on a network. DFS makes files easier for users to locate, and increases availability by maintaining multiple file copies across distributed servers.

Encrypting File System

Windows 7 and Windows Server 2008 R2 continue to provide security of data on the hard disk by encrypting files. This data remains encrypted even when backed up or archived. The Encrypting File System (EFS) runs as an integrated system service making it easy to manage, difficult to attack, and transparent to the user. The encryption and decryption processes are transparent to the user, once files are marked for encryption. Performance enhancements in Windows 7 and Windows Server 2008 R2 include support for encrypting the paging file, and storage of user EFS keys on smart cards.

EFS Multi-user Support

Windows 7 and Windows Server 2008 R2 support file sharing between multiple users of an individual encrypted data file. Encrypted file sharing is a useful and easy way to enable collaboration without having to share private keys among users.

Event Logging Infrastructure

Windows 7 and Windows Server 2008 R2 introduce improvements to the event logging infrastructure that make the platform easier to manage, monitor, and provide better information for troubleshooting. Many components that stored logging information in text files in previous versions are now able to add events to the event log. With event forwarding, administrators can centrally manage events from remote computers on the network, making it easier to identify problems and to correlate problems that affect multiple computers. Additionally, the Event Viewer application allows users to create custom views of audit data, to easily associate events with tasks, and to remotely view logs from other computers.

Fault-Tolerant Process Model and Kernel-Mode Web Driver

With IIS, web traffic requests are passed directly from the network stack to a kernel-mode Web driver, HTTP.SYS. The AFD.SYS driver and Winsock 2.0 layer do not play a role. HTTP.SYS examines the request, determining if it can be satisfied from the driver’s own cache. If so, the requested content is immediately returned without a context switch from kernel mode to user mode. When the kernel-mode Web driver cannot satisfy a request from its cache, HTTP.SYS passes the request across the kernel/user boundary directly to a worker process for servicing. The architecture of IIS significantly improves Web server stability because a single faulty application running on the Web server cannot bring down other applications on the same server. A worker process (see IIS Web Service below) that is servicing the faulty application can simply be recycled without affecting other worker processes.

File Replication Service

File Replication Service (FRS) is a technology that replicates files and folders (and their security and other attributes) stored in the System Volume (SYSVOL) shared folder between domain controllers and between Distributed File System (DFS) shared folders. When FRS detects that a change has been made to a file or folder within a replicated shared folder, FRS replicates the updated file or folder to other servers. Because FRS is a multi-master replication service, any server that participates in the replication of a shared folder can generate changes. In addition, FRS can resolve file and folder conflicts to make data consistent among servers.

Forest Trust

Forest trust is a type of Windows trust for managing the security relationship between two Active Directory (AD) forests. This feature enables the trusting forest to enforce constraints on which security principal names it trusts other forests to authenticate. This new trust type that allows all domains in one forest to (transitively) trust all domains in another forest, via a single trust link between the two forest root domains. Cross-forest authentication enables secure access to resources when the user account is in one forest and the computer account is in another forest. This feature allows users to securely access resources in other forests, using either Kerberos or NTLM[18], without sacrificing the single sign-on benefits of having only one user Identification (ID) and password maintained in the user's home forest.

Globally Unique Identifier (GUID) Partition Table (GPT)

Windows 7 and Windows Server 2008 R2 support a disk partitioning mechanism, the GUID Partition Table (GPT). Unlike master boot record partitioned disks, GPT allows data critical to platform operation to be located in partitions rather than unpartitioned or hidden sectors. In addition, GPT partitioned disks provide improved data structure integrity by offering redundant primary and backup partition tables.

Group Policy

Windows 7 and Windows Server 2008 R2 Group policy allows central management of collections of users, computers, applications, and network resources instead of managing entities on a one-by-one basis. Integration with AD delivers granular and flexible control. It permits authorized administrators to define customized rules about virtually every facet of a user's computer environment such as security, user rights, desktop settings, applications, and resources, minimizing the likelihood of misconfiguration. Windows 7 and Windows Server 2008 R2 add numerous additional policy settings to those available in previous versions of the operating system.

Upon installation, Windows 7 and Windows Server 2008 R2 offer groups that are pre-configured with specific user rights and/or privileges. These groups are referred to as “built-in groups.” The Windows 7 and Windows Server 2008 R2 built-in groups fall into three (3) categories: built-in local groups (e.g., Administrator, Backup Operator); built-in domain local groups (e.g., Administrator, Account Operator);

[18] NTLM is Windows Challenge / Response described below.

and built-in global groups (e.g. Enterprise Administrator, Domain Administrator). The authorized administrator can conveniently take advantage of these built-in groups by assigning these groups to specific user accounts allowing users to gain the rights and/or privileges associated with these groups.

Hardware Data Execution Prevention

64-bit hardware support adds a set of Data Execution Prevention (DEP) security checks to the TOE. These checks, known as hardware-enforced DEP, are designed to block malicious code that takes advantage of exception-handling mechanisms by intercepting attempts to execute code in memory that is marked for data only. This hardware protection feature is present in all 64-bit hardware architectures in the evaluated configuration.

While not available for 32-bit hardware architectures, due to hardware limitations, the only limitation is that application programs are not afforded additional protection from potential programming errors that might be exploitable by malicious users.

High Throughput and Bandwidth Utilization

Windows 7 and Windows Server 2008 R2 include many enhancements to those core OS functions that are used to manipulate and manage system resources. Because the efficiency with which system resources are managed affects all server workloads, the benefits resulting from these changes are not limited to any one workload but instead have a broad, positive impact on performance and scalability. Most server workloads have some component of disk I/O and/or network I/O. Both types of I/O require processor cycles and memory, so the optimizations in Windows Server 2008 R2 that improve the efficiency with which disk I/O and network I/O is processed leave more system resources available to support other components of a workload.

IIS Web Service

An IIS worker process is an application that runs in user mode. Its typical roles include processing requests to return a static page, invoking an Internet Server API (ISAPI) extension or filter, or running an application specific handler. A worker process is physically implemented as an executable file named “W3wp.exe” and is controlled by World-Wide Web (WWW) Service Administration and Monitoring. By default, worker processes run as Network Service, which has the least system resource access that is compatible with the functionality required. Worker processes use kernel-mode HTTP.SYS IIS driver to send requests and receive responses over HTTP. Depending on how IIS is configured, there can be multiple worker processes running, serving different Web applications concurrently. This design separates applications by process boundary, and it helps achieve maximum Web server reliability and security.

Integrated IPSec Support

Windows 7 and Windows Server 2008 R2 include identical IPSec support for both IPv4 and IPv6. Full support for Internet Key Exchange (IKE) and data encryption is provided for both IP stacks. IPSec configuration is integrated with the Windows Firewall with Advanced Security MMC snap-in to improve manageability and reduce the likelihood of conflicting firewall and IPSec rules.

Internet Connection Sharing

Internet Connection Sharing (ICS) is intended for use in a scenario where the ICS host computer directs network communication between two networks where one network is typically a more private LAN while the other is typically a wide area network. The ICS host computer needs two network connections. The LAN connection, automatically created by installing a network adapter, connects to the computers on the LAN. The other connection connects the LAN to the Wide Area Network (WAN). As a result, the shared connection connects computers on the LAN to the WAN.

IPv6

Windows 7 and Windows Server 2008 R2 provide a dual IP stack in which IPv4 and IPv6 are implemented alongside each other and share a common IP transport (including TCP and UDP) IPv6 is enabled by default and supports numerous enhancements including a GUI based configuration, improvements to Teredo (an IPv6 transition technology), generation of interface IDs, a DHCPv6 client that support stateful address auto configuration, and for Windows Server 2008 R2, a DHCPv6 capable server.

Job Object API

The Windows 7 and Windows Server 2008 R2 Job Object API, with its ability to specify processor affinity, establish time limits, control process priorities, and limit memory utilization for a group of related processes, enabling an application to manage and control dependent system resources. This additional level of control means the Job Object API can prevent an application from negatively impacting overall system scalability.

Kerberos Authentication Support

Full support for Kerberos Version 5 (v5) protocol Windows 7 and Windows Server 2008 R2 provides fast, single sign-on to Windows 7 and Windows Server 2008 R2 based enterprise resources. It is used to support Transitive Domain Trust to reduce the number of trust relationships required to manage users and resources between Windows domains.

Kernel Debug Management

The Kernel Debugger subcomponent supports authorized users to debug running processes in the Windows 7 and Windows Server 2008 R2 TOE by allowing them to attach a debugger to a running process via a kernel object, the “Debug Object”. The Kernel Debugger associates resources implemented by other kernel-mode subcomponents and wraps them in a debug object that can then be manipulated to provide information about the system that was previously unavailable without the aid of an external debugger.

Kernel Transaction Manager

Windows includes a transaction engine that enables applications to use atomic transactions on resources to facilitate improved error recovery. This transaction engine allows transactional resource managers such as the NT File System (NTFS) and the Configuration Manager (e.g., the registry) to coordinate their updates for a specific set of changes made by an application. NTFS uses an extension to support transactions called TxF. The Configuration Manager uses a similar extension called TxR. These kernel-mode resource managers work with the kernel transaction manager to coordinate the transaction state, just as user-mode resource managers use the Distributed Transaction Coordinator to coordinate transaction state across multiple user-mode resource managers.

Mandatory Integrity Control

In addition to Discretionary Access Control (DAC), Windows provides Mandatory Integrity Control (MIC). MIC uses integrity levels and a mandatory policy to evaluate access. Processes and securable objects (e.g., files) are assigned integrity levels that determine their levels of protection or access.

As an integrity policy, a process with a lower integrity level (e.g., low) cannot write to an object with a higher integrity level (e.g., medium), even if that object's DAC policy allows write access. On the other hand, processes can access objects that have an integrity level lower than or equal to their own integrity level. In addition, the MIC policy addresses read and execute accesses, and can be configured to restrict a process with a lower integrity level from reading and/or executing objects with a higher integrity level.

The integrity labels defined in Windows are:

- Untrusted: Used by processes started by the Anonymous group;
- Low: Used by protected mode IE, blocks write access to most objects (such as files and registry keys) on the system;
- Medium: Normal applications being launched while user account control (UAC) is enabled;
- High: Applications launched through administrator elevation when UAC is enabled, or normal applications if UAC is disabled; and
- System: Services and other system-level applications (such as WinLogon).

Microsoft Management Console

Microsoft Management Console (MMC) unifies and simplifies system management tasks through a central, customizable console that allows control, monitoring, and administration of widespread network resources. MMC 3.0 provides a new add or remove snap-ins dialog box, improved error handling, and an action pane that provides context sensitive access to features based on the currently selected items in the tree or results pane.

Multi-master Replication

AD uses multi-master replication to ensure high scalability and availability in distributed network configurations. "Multi-master" means that each directory replica in the domain is a peer of all other replicas; changes can be made to any replica and will be reflected across all of them.

Network Address Translation

Network Address Translation (NAT) hides internally managed IP addresses from external networks by translating private internal addresses to public external addresses. This translation reduces IP address registration costs by using private IP addresses internally, which are translated to a small number of registered IP addresses externally. NAT also hides the internal network structure, reducing the risk of attacks against internal systems. The Windows 7 and Windows Server 2008 R2 TOE IPSec implementation works transparently with NAT without interoperability issues.

Network Bridge

The Network Bridge feature provides an easy and inexpensive way to connect LAN segments. Through Network Bridge, users can bridge connections among different computers and devices on their network, even when they connect to the network through different methods.

Plug and Play

Plug and Play technology combines hardware and software support in such a way that the Windows 7 and Windows Server 2008 R2 TOE can recognize and adapt to hardware configuration changes automatically, without user intervention and or restarting the computer.

Processor Run Time Power Management

For each family of processors supported by the Windows 7 and Windows Server 2008 R2 TOE, an abstraction of issues dealing with processor frequency, voltage, microcode, temperature, idle handling, starting, stopping and initialization is defined. The TOE uses this abstraction to manage the power management aspect of the processors.

Protocol Transition

In Windows Server 2008 R2 TOE, the Kerberos protocol transition mechanism allows a service to transition to a Kerberos-based identity for the user without knowing the user’s password and without the user having to authenticate using Kerberos. For example, a user’s network logon to a server can be authenticated using the NTLM or Digest protocols, the server and then obtains a Kerberos ticket for the user’s Windows identity, subject to system policy.

Public Key Certificate Issuing and Management Service

The Windows Server 2008 R2 Certificate Server issues and manages public key certificates for the following Windows 7 and Windows Server 2008 R2 TOE services: digital signatures, software code signing, TLS/SSL authentication for Web traffic, IPSec, Smart card logon, EFS user and recovery certificates.

Remote Storage Service

Remote Storage uses criteria specified by an authorized user to automatically copy little-used files to removable media. If hard-disk space drops below specified levels, Remote Storage removes the (cached) file content from the disk. If the file is needed later, the content is automatically recalled from storage. If the media is not present, a dialog box prompting to load the media is displayed at the server's console.

Secure Network Communications

Windows 7 and Windows Server 2008 R2 support end-to-end encrypted communications across network using the IPSec standard. It protects sensitive internal communications from intentional or accidental viewing. AD provides central policy control for its use to make it deployable.

Smart Card Support for Authentication

Smart Card technology is fully integrated into the Windows 7 and Windows Server 2008 R2 TOE, and is an important component of the operating system's Public Key Infrastructure (PKI) security feature. The smart card serves as a secure store for public and private keys and as a cryptographic engine for performing a digital signature or key-exchange operation. Smart card technology allows Windows 7 and Windows Server 2008 R2 TOE to authenticate users by using the private and public key information stored on a card. The Smart Card subsystem on the Windows 7 and Windows Server 2008 R2 TOE supports industry standard Personal Computer/Smart Card (PC/SC)–compliant cards and readers, and provides drivers for commercially available Plug and Play smart card readers. Smart card readers attach to standard peripheral interfaces, such as Universal Serial Bus (USB). The Windows 7 and Windows Server 2008 R2 TOE detects Plug and Play-compliant smart card readers and installs them using the Add Hardware wizard.

Super Fetch

Windows 7 and Windows Server 2008 R2 include a Super Fetch feature that allows Windows 7 and Windows Server 2008 R2 to monitor application usage so that it can predict future application requirements and pre-load common or regularly used executable code into the memory cache to improve their perceived load times.

Support for Security Standards

Windows 7 and Windows Server 2008 R2 build secure network sites using the latest standards, including 128-bit and 256-bit SSL/TLS, IPSec ,and Kerberos v5 authentication.

URL-Based authorization

This authorization mechanism enables businesses to control access to applications exposed through the Web by restricting user access to URLs. For example, one user may be restricted from access to certain applications, whereas another user can be allowed to execute other applications.

User Account Control

User Account Control (UAC) (alternately known as LUA – Least Privilege User Access) enables users to perform common tasks as non-administrators, called standard users, and as administrators without having to switch users, log off, or use the Run As command. A standard user account is synonymous with a user account in Windows 7 and Windows Server 2008 R2. User accounts that are members of the local Administrators group will run most applications as a standard user.

When an administrator logs on to a computer running Windows 7 or Windows Server 2008 R2, the user is assigned two separate access tokens. Access tokens, which contain a user's access control data, group membership and authorization data, are used by Windows to control what resources and tasks the user can access. In early versions of Windows, an administrator account received only one access token, which included data to grant the user access to all Windows resources. This access control model did not include any checks to ensure that users truly wanted to perform a task that required their administrative access token.

When an administrator logs on to a computer running Windows 7 or Windows Server 2008 R2, the user’s full administrator access token is split into two access tokens: a full administrator access token and a standard user access token. During the logon process, authorization and access control components that identify an administrator are removed, resulting in a standard user access token. The standard user access token is then used to start the Widows desktop process. Because all applications inherit their access control data from the initial launch of the desktop, they all run as a standard user as well.

After an administrator logs on, the full administrator access token is not invoked until the user attempts to perform an administrative task at which point the user will be interactively prompted to confirm this access escalation.

Virtual Disk Service

Virtual Disk Service (VDS) provides a set of utilities for managing the hardware disks. VDS implements a single, uniform interface for managing disks. Each hardware storage vendor writes a VDS provider that translates the general purpose VDS APIs into specific instructions for their hardware. Windows 7 and Windows Server 2008 R2 include VDS providers for basic and dynamic disks.

Virtualization

The Hyper-V role in Windows Server 2008 R2 (and its predecessor Windows Server 2008) provides software infrastructure and basic management tools to create and manage a virtualized server computing environment.

Volume Shadow Copy Service

Volume Shadow Copy Service (VSS) coordinates shadow copies for applications and target NTFS volumes in a point-in-time copy. This feature was enhanced in Windows 7 and Windows Server 2008 R2 bringing support for the feature to all systems. Volume snapshots are automatically created, typically once per day, and can be accessed through the Windows Explorer file and folder properties dialogs using the same interface used by Shadow Copies for Shared Folders. This enables users to view, restore, or copy old versions of files and directories that might have accidentally been modified or deleted.

Web Document Authoring and Versioning (WebDAV) Redirector

WebDAV redirector allows files stored in web folders to be encrypted with EFS. When a client maps a drive to a WebDAV access point on a remote server, files may be encrypted locally on the client and then transmitted as a encrypted file to the WebDAV server using an HyperText Transfer Protocol (HTTP) “PUT” command. Similarly, encrypted files downloaded to a client are transmitted as encrypted files using an HTTP “GET” command and decrypted locally on the client.

Web Site Permissions

An authorized user can configure web site's access permissions for specific sites, directories, and files. Web Site permissions are not meant to be used in place of NTFS permissions. Instead, they are used with NTFS permissions to strengthen the security of specific Web site content maintained by the IIS web server of the Windows Server 2008 R2 TOE. Unlike NTFS permissions, Web site permissions affect everyone who tries to access the configured Web sites. If Web permissions conflict with NTFS permissions for a directory or file, the more restrictive settings are applied.

Windows File Protection

The Windows File Protection technology prevents core system files from being overwritten by application installs. In the event a file is overwritten, Windows File Protection will replace that file with the correct version. Windows 7 and Windows Server 2008 R2 identify device drivers that have passed the Windows Hardware Quality Labs test and warns users if they are about to install an uncertified driver.

Windows Firewall (previously known as Internet Connection Firewall (ICF))

Windows Firewall is a stateful firewall that drops unsolicited incoming traffic which does not correspond to either (1) traffic sent in response to a request of the computer (solicited traffic) or (2) unsolicited traffic that has been specified as allowed (excepted traffic). Windows Firewall provides a level of protection from malicious users and programs that rely on unsolicited incoming traffic to attack computers. Windows Firewall supports IPv4 and IPv6. The firewall drivers (for IPv4 and for IPv6 respectively) have a static rule called a boot-time policy to perform stateful filtering. This allows the Windows 7 and Windows Server 2008 R2 to perform basic networking tasks such as DNS and DHCP and communicate with a DC to obtain policy. Once the firewall service is running, it will load and apply the runtime policy and remove the boot-time filters.

Window Manager

The Window Manager is implemented in kernel mode. It provides a machine independent graphical Application Programming Interface (API) for applications to control printing and window graphics, by providing a way to display information and receiving user input. Users interact with the application thorough graphical features. They can control applications by choosing menu commands. They can provide input using the mouse, keyboard, and other devices. They receive information from resources such as bitmaps, carets, cursors, and icons. The Window Manager exports two protected object types: Window station objects and Desktop Objects. Each is an object with a Discretionary Access Control List (DACL) that is used to control access to it.

Windows Installer Service

The Windows Installer Service enables customers to better address corporate deployment and provide a standard format for component management. The installer supports advertisement of applications and features according to the operating system settings. It can install multiple updates with a single transaction that integrates installation progress, rollback, and reboots. It can apply patches in a constant order regardless of the order that the patches are provided to the system. Patches installed with the Windows Installer Service can be uninstalled in any order to leave the state of the product the same as if the patch was never installed. Patching using Windows Installer Service only updates files affected by the patch and can be significantly faster than earlier installer versions. Accounts with administrator privileges can use Windows Installer Service functions to query and inventory product, feature, component and patch information and to read, edit and replace installer source lists for network, URL and media sources. Administrators can enumerate across user and install contexts and manage source lists from an external process.

Windows Management Instrumentation

Windows Management Instrumentation (WMI) is a uniform model through which management data from any source can be managed in a standard way. WMI provides this for software, such as applications, while WMI extensions for the Windows Driver Model (WDM) provide this for hardware or hardware device drivers.

“Winsock2” Installable File System Layer Driver

The “Winsock2” Installable File System (IFS) Layer Driver is a transport layer driver that emulates file handles for Windows Socket service providers for which a socket handle is not an IFS handle. As a result, Windows Sockets architecture accommodates service providers whose socket handles are not IFS objects. Applications can use Win32 file I/O calls with the handle without any knowledge about the network aspects.

Windows Security Center Service (WSC)

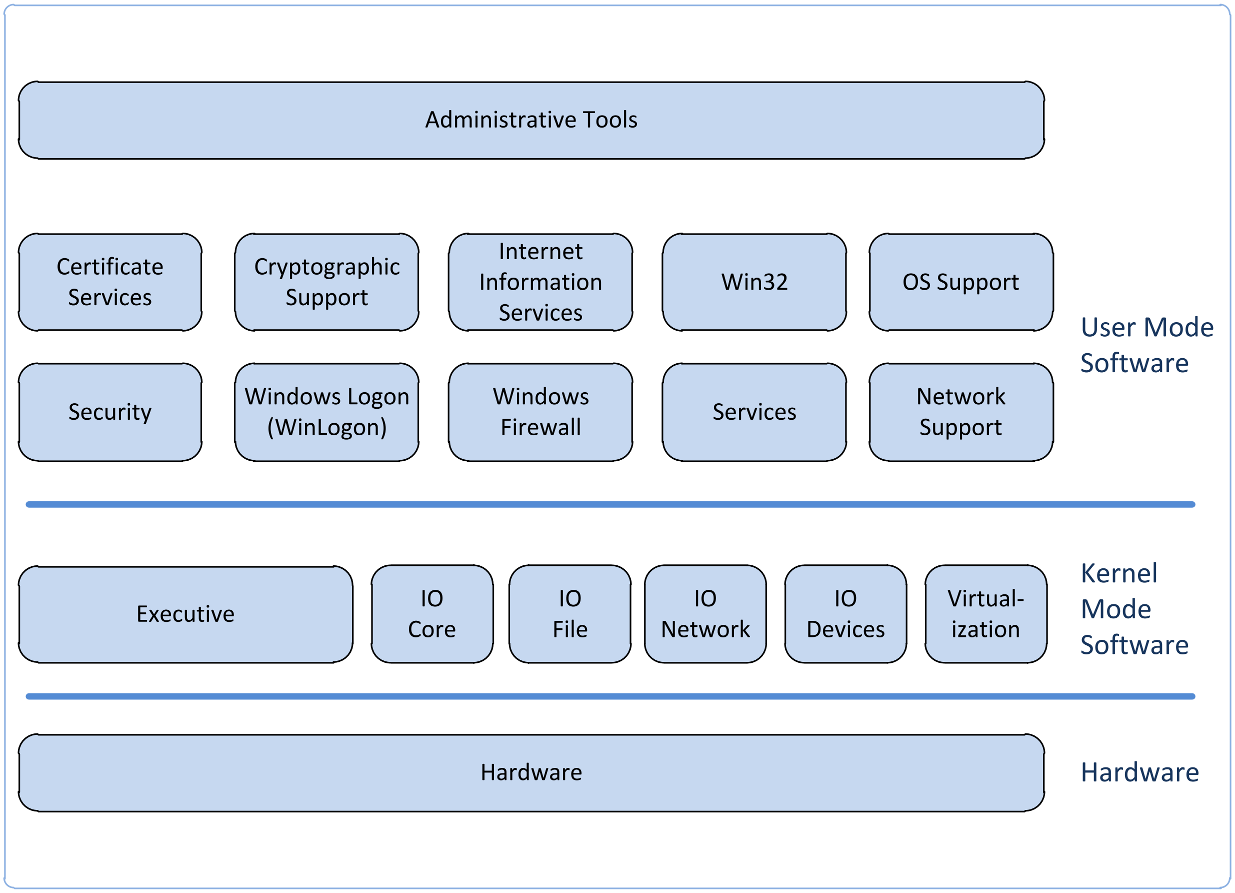
The Windows Security Center Service (WSC) is a service that monitors, among other things, the status of Windows Firewall running on the Windows 7 and Windows Server 2008 R2. It also provides the logged-on interactive user certain visual notifications when it detects that the status of Windows Firewall has changed.

2.4 Security Environment and TOE Boundary

The TOE includes both physical and logical boundaries. Its operational environment is that of a networked environment.

2.4.1 Logical Boundaries

The diagram below depicts components and subcomponents of Windows 7 and Windows Server 2008 R2 that comprise the TOE. The components/subcomponents are large portions of the Windows 7 and Windows Server 2008 R2 operating system, and generally fall along process boundaries and a few major subdivisions of the kernel mode software.



- Administrative Tools Module
    - ○ Administrative Tools Component: This component represents the range of tools available to manage the security properties of the TSF.
- Certificate Services Module
    - ○ Certificate Server Component: This component provides services related to issuing and managing public key certificates (e.g. X.509 certificates).
- Windows Firewall Module

  - Windows Firewall Component: This component provides services related to information flow control.
- Hardware Module
  - Hardware Component: This component includes all hardware used by the TSF to include the processor(s), motherboard and associated chip sets, controllers, and I/O devices.
- Kernel Software Module
  - Executive Component: This is the kernel-mode software that provides core OS services to include memory management, process management, and inter-process communication. This component implements all the non-I/O TSF interfaces for the kernel-mode.
  - I/O System: This is the kernel-mode software that implements all I/O related services, as well as all driver-related services. The I/O System is further divided into:
    - I/O Core Component
    - I/O File Component
    - I/O Network Component
    - I/O Devices Component
  - Virtualization: This is kernel-mode software that supports server virtualization as well as driver-related services to provide a virtualized set of device drivers to operating systems running on a guest partition.
- [Miscellaneous] OS Support Module
  - OS Support Component: This component is a set of processes that provide various other OS support functions and services.
- Network Support Module
  - Network Support Component: This component contains various support services for RPC, COM, and other network services.
- Security Module
  - Security Component: This component includes all security management services and functions.
- Services Module
  - Services Component: This is the component that provides many system services as well as the service controller.
- Internet Information Services Module
  - IIS Component: This component provides services related to Web/HTTP requests.
- Win32 Module
  - Win32 Component: This component provides various support services for Win32 applications and the command console application.
- WinLogon Module
  - WinLogon Component: This component provides various interactive logon services to include interactive authentication, trusted path, session management and locking.
- Cryptographic Support Module

- Cryptographic Support Component: This component provides cryptographic services for use by the kernel and other components in a manner that keeps them distinct from other components of the TOE.

These components are further refined in **Appendix B—TOE Component Decomposition**.

2.4.2 Physical Boundaries

Physically, each TOE workstation or server consists of an x86, x64, or IA64 machine or equivalent processor (from the Intel Celeron, Intel Pentium, Intel Core 2, Intel Itanium, AMD Sempron, AMD Athlon, or AMD Phenom processor families) with up to four (4) CPUs for a standard Server product, up to eight (8) CPUs for the Enterprise Server product, and up to 32 CPUs for the Datacenter product. A set of devices may be attached and they are listed as follows:

- Display Monitor
- Keyboard
- Mouse
- CD-ROM Drive
- Fixed Disk Drives
- Printer
- Audio Adaptor
- Network Adaptor
- Smart Card Reader
- TPM

The TOE does not include any physical network components between network adaptors of a connection. The ST assumes that any network connections, equipment, and cables are appropriately protected in the TOE security environment.

2.5 TOE Security Services

This section summarizes the security services provided by the TOE:

- **Security Audit:** Windows 7 and Windows Server 2008 R2 have the ability to collect audit data, review audit logs, protect audit logs from overflow, and restrict access to audit logs. Audit information generated by the system includes the date and time of the event, the user identity that caused the event to be generated, and other event specific data. Authorized administrators can review audit logs and have the ability to search and sort audit records. Authorized Administrators can also configure the audit system to include or exclude potentially auditable events to be audited based on a wide range of characteristics.
- **Identification and Authentication (I&A):** Windows 7 and Windows Server 2008 R2 require each user to be identified and authenticated (using password or smart card) prior to performing any functions. An interactive user invokes a trusted path in order to protect his I&A information. Windows 7 and Windows Server 2008 R2 maintain databases of accounts including their identities, authentication information, group associations, and privilege and logon rights

associations. Windows 7 and Windows Server 2008 R2 include a set of account policy functions that include the ability to define the minimum password length, the number of failed logon attempts, the duration of lockout, and password age.

- **Security Management**: Windows 7 and Windows Server 2008 R2 include a number of functions to manage policy implementation. Policy management is controlled through a combination of access control, membership in administrator groups, and privileges.
- **User Data Protection**: Windows 7 and Windows Server 2008 R2 protect user data by enforcing several access control policies (Discretionary Access Control, Mandatory Integrity Control, Encrypting File System, WEBUSER and web content provider access control) and several information flow policies (IPSec filter information flow control, Windows Firewall); and, object and subject residual information protection. Windows 7 and Windows Server 2008 R2 use access control methods to allow or deny access to objects, such as files, directory entries, printers, and web content. Windows 7 and Windows Server 2008 R2 uses information flow control methods to control the flow of IP traffic and packets. Windows authorizes access to these resource objects through the use of SDs (which are sets of information identifying users and their specific access to resource objects), web permissions, IP filters, and port mapping rules. Windows 7 and Windows Server 2008 R2 also protect user data by ensuring that resources exported to user-mode processes do not have any residual information.
- **Cryptographic Protection:** Windows 7 and Windows Server 2008 R2 provide FIPS-140-2 validated cryptographic functions that support encryption/decryption, cryptographic signatures, cryptographic hashing, cryptographic key agreement, and random number generation. The TOE additionally provides support for public keys, credential management and certificate validation functions and provides support for the National Security Agency's Suite B cryptographic algorithms. The TOE also provides extensive auditing support of cryptopgraphic operations, support for replaceable random number generators, and a key isolation service designed to limit the potential exposure of secret and private keys. In addition to supporting its own security functions with cryptographic support, the TOE offers access to the cryptographic support functions for user application programs.
- **Protection of TOE Security Functions**: Windows 7 and Windows Server 2008 R2 provide a number of features to ensure the protection of TOE security functions. Windows 7 and Windows Server 2008 R2 protect against unauthorized data disclosure and modification by using a suite of Internet standard protocols including IPSec and ISAKMP. Windows 7 and Windows Server 2008 R2 ensure process isolation security for all processes through private virtual address spaces, execution context, and security context. The Windows 7 and Windows Server 2008 R2 data structures defining process address space, execution context, memory protection, and security context are stored in protected kernel-mode memory. The Windows 7 and Windows Server 2008 R2 BitLocker features can be used to protect fixed and removable USB storage volumes. The Windows 7 and Windows Server 2008 R2 Network Access Protection feature can be used to limit access to network resources depending on the measured "health" of clients based on their security settings, installed applications, etc. The Windows 7 and Windows Server

2008 R2 also include some self-testing features that ensure the integrity executable TSF image and its cryptographic functions.

- **Resource Utilization**: Windows 7 and Windows Server 2008 R2 can limit the amount of disk space that can be used by an identified user or group on a specific disk volume. Each volume has a set of properties that can be changed only by a member of the administrator group. These properties allow an authorized administrator to enable quota management, specify quota thresholds, and select actions when quotas are exceeded.
- **Session Locking**: Windows 7 and Windows Server 2008 R2 provide the ability for a user to lock their session either immediately or after a defined interval. Windows constantly monitors the mouse and keyboard for activity and locks the workstation after a set period of inactivity. Windows 7 and Windows Server 2008 R2 allow an authorized administrator to configure the system to display a logon banner before the logon dialogue.

3 Security Problem Definition

The TOE security problem definition consists of the threats to security, organizational security policies, and usage assumptions as they relate to Windows 7 and Windows Server 2008 R2. The assumptions, threats, and policies are derived from the GPOSPP.

3.1 Threats to Security

Table 3-1 presents known or presumed threats to protected resources that are addressed by Windows 7 and Windows Server 2008 R2.

Table 3-1 Threats Addressed by Windows 7 and Windows Server 2008 R2

Threat	Description
T.ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.ADMIN_ROGUE	An authorized administrator’s intentions may become malicious resulting in user or TSF data being compromised.
T.AUDIT_COMPROMISE	A malicious user or process may view audit records, cause audit records to be lost or modified, or prevent future audit records from being recorded, thus masking a user’s action.
T.CRYPTO_COMPROMISE	A malicious user or process may cause key, data or executable code associated with the cryptographic functionality to be inappropriately accessed (viewed, modified, or deleted), thus compromising the cryptographic mechanisms and the data protected by those mechanisms.
T.MASQUERADE	A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources.
T.OPERATIONAL_ERRORS	While the TOE is operational, changes to the TOE may cause it to enter a configuration that is not able to enforce the security policies of the TOE.
T.REPLAY	A user may gain inappropriate access to the TOE by replaying authentication information, or may cause the TOE to be inappropriately configured by replaying TSF data or security attributes.
T.RESIDUAL_DATA	A user or process may gain unauthorized access to data through reallocation of TOE resources from one user or process to another.
T.RESOURCE_EXHAUSTION	A malicious process or user may block others from system resources (i.e., persistent storage) via a resource exhaustion denial of service attack.
T.TSF_COMPROMISE	A malicious user or process may cause TSF data or executable code to be inappropriately accessed (viewed, modified or deleted).
T.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
T.UNAUTHORIZED_ACCESS	A user may gain unauthorized access (view, modify, delete) to user

Threat	Description
	data.
T.UNIDENTIFIED_ACTIONS	The administrator may fail to notice potential security violations, thus preventing the administrator from taking action against a possible security violation.
T.UNKNOWN_STATE	When the TOE is initially started or restarted after a failure, the security state of the TOE may be unknown.

3.2 Organizational Security Policies

Table 3-2 describes organizational security policies that are addressed by Windows 7 and Windows Server 2008 R2.

Table 3-2 Organizational Security Policies

Security Policy	Description
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.
P.ACCOUNTABILITY	The users of the TOE shall be held accountable for their actions within the TOE.
P.AUTHORIZATION	The TOE shall limit the extent of each user's abilities in accordance with the TSP.
P.AUTHORIZED_USERS	Only those users who have been authorized to access the information within the TOE may access the TOE.
P.CRYPTOGRAPHY	The TOE shall use NIST FIPS validated cryptography as a baseline for key management (i.e., generation and destruction) and for cryptographic operations (i.e., encryption, decryption, signature, hashing, and random number generation services).
P.I_AND_A	All users must be identified and authenticated prior to accessing any controlled resources with the exception of public objects.
P.NEED_TO_KNOW	The TOE must limit the access to data in protected resources to those authorized users who have a need to know that data.
P.ROLES	The TOE shall provide multiple administrative roles for secure administration of the TOE. These roles shall be separate and distinct from each other.
P.TRACE	The TOE shall provide the ability to review the actions of individual users.
P.TRUSTED_RECOVERY	Procedures and/or mechanisms shall be provided to assure that, after a TOE failure or other discontinuity, recovery without a protection compromise is obtained.

3.3 Secure Usage Assumptions

Table 3-3 describes the security aspects of the environment in which Windows 7 and Windows Server 2008 R2 is intended to be used.

Table 3-3 Secure Usage Assumptions

Assumption	Description
A.PHYSICAL	It is assumed that the IT environment provides the TOE with appropriate physical security, commensurate with the value of the IT assets protected by the TOE.

4 Security Objectives

This section defines the security objectives of Windows 7 and Windows Server 2008 R2 and its supporting environment. Security objectives reflect the stated intent to counter identified threats, comply with any organizational security policies identified, or address identified assumptions. All of the identified threats, organizational policies, and assumptions are addressed under one of the categories below.

4.1 TOE Security Objectives

Table 4-1 describes the Windows 7 and Windows Server 2008 R2 security objectives.

Table 4-1 TOE Security Objectives

Security Objective	Description
O.ACCESS	The TOE will ensure that users gain only authorized access to it and to resources that it controls.
O.ACCESS_HISTORY	The TOE will display information (to authorized users) related to previous attempts to establish a session.
O.ADMIN_ROLE	The TOE will provide administrator roles to isolate administrative actions.
O.AUDIT_GENERATION	The TOE will provide the capability to detect and create records of security relevant events associated with users.
O.AUDIT_PROTECTION	The TOE will provide the capability to protect audit information.
O.AUDIT_REVIEW	The TOE will provide the capability to selectively view audit information and alert the administrator of identified potential security violations.
O.CORRECT_TSF_OPERATION	The TOE will provide a capability to test the TSF to ensure the correct operation of the TSF in its operational environment.
O.CRYPTOGRAPHIC_SERVICES	The TOE will make encryption services available to authorized users and/or user applications.
O.DISCRETIONARY_ACCESS	The TOE will control access to resources based upon the identity of users and groups of users.
O.DISCRETIONARY_USER_CONTROL	The TOE will allow authorized users to specify which resources may be accessed by which users and groups of users.
O.DISPLAY_BANNER	The TOE will display an advisory warning regarding use of the TOE.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the authorized administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.PROTECT	The TOE will provide mechanisms to protect user data and resources.
O.RECOVERY	Procedures and/or mechanisms will be provided to assure that recovery is obtained without a protection compromise, such as

Security Objective	Description
	from system failure or discontinuity.
O.RESIDUAL_INFORMATION	The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated.
O.RESOURCE_EXHAUSTION	The TOE shall provide mechanisms that mitigate user attempts to exhaust persistent storage.
O.DOMAIN_ISOLATION	The TOE will maintain a domain for its own execution that protects itself and its resources from external interference, tampering, or unauthorized disclosure.
O.TSF_CRYPTOGRAPHIC_INTEGRITY	The TOE will provide cryptographic integrity mechanisms for TSF data while in transit to remote parts of the TOE.
O.USER_AUTHENTICATION	The TOE will verify the claimed identity of users.
O.USER_IDENTIFICATION	The TOE will uniquely identify users.

4.2 Security Objectives for the Operational Environment

The TOE is assumed to be complete and self-contained and, as such, is not dependent upon any other products to perform properly. However, certain objectives with respect to the general operating environment must be met. **Table 4-2** describes the Security Objectives for the Operational Environment.

Table 4-2 Security Objectives for the Operational Environment

Security Objective	Description
OE.PHYSICAL	Physical security will be provided for the TOE by the IT environment, commensurate with the value of the IT assets protected by the TOE.

5 Security Requirements

The section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) for the TOE. The requirements in this section have been drawn from the US Government Protection Profile for General-Purpose Operating Systems in a Networked environment (GPOSPP) (version .7, 10 August 2009), Common Criteria, or are defined in the following section.

Conventions:

Where requirements are drawn from the GPOSPP, the requirements are copied verbatim (except for some changes to required identifiers to reflect the iteration convention of this document) from that Protection Profile and only operations performed in this Security Target are identified.

Where requirements are drawn from the Common Criteria (and are not found in the GPOSPP), the requirements are copied verbatim (except for some changes to required identifiers to reflect the iteration convention of this document) and the operations performed in this Security Target are identified.

Requirement defined in this Security Target naturally have no identified operations.

Where applicable the following conventions are used to identify operations:

- **Iteration**: Iterated requirements (components and elements) are identified with letter following the base component identifier. For example, iterations of FCS_COP.1 are identified in a manner similar to FCS_COP.1a (for the component) and FCS_COP.1a.1 (for the elements).
- **Assignment**: Assignments are identified in brackets and bold (e.g., **[assigned value]**).
- **Selection**: Selections are identified in brackets, bold, and italics (e.g., ***[selected value]***).
    - Assignments within selections are identified using the previous conventions, except that the assigned value would also be italicized and extra brackets would occur (e.g., ***[selected value [assigned value]]***).
- **Refinement**: Refinements are identified using bold text (e.g., **added text**) for additions and strike-through text (e.g., ~~deleted text~~) for deletions.

5.1 Extended Components Definitions

The following extended components are identified in this Security Target:

- FCO_NRO_CIMC.3: Enforced Proof of Origin and Verification of Origin
- FCO_NRO_CIMC.4: Advanced Verification of Origin
- *FCS_BCM_EXT.1: Baseline Cryptographic Module*
- *FCS_COA_EXT.1: Cryptographic Operations Availability*
- *FCS_RBG_EXT.1: Random Number Generation*
- FDP_ACF_CIMC.2: User Private Key Confidentiality Protection

- FDP_ACF_CIMC.3: User Secret Key Confidentiality Protection
- FDP_CIMC_CER.1: Certificate Generation
- FDP_CIMC_CRL.1: Certificate Revocation List Validation
- FDP_CIMC_CSE.1: Certificate Status Export
- FDP_CIMC_OCSP.1: OCSP Basic Response Validation
- FDP_ETC_CIMC.5: Extended User Private and Secret Key Export
- *FIA_AFL_EXT.1: Authentication Failure Handling*
- FMT_MOF_CIMC.3: Extended Certificate Profile Management
- FMT_MOF_CIMC.5: Extended Certificate Revocation List Profile Management
- FMT_MOF_CIMC.6: OCSP Profile Management
- FMT_MTD_CIMC.4: TSF Private Key Confidentiality Protection
- FMT_MTD_CIMC.5: TSF Secret Key Confidentiality Protection
- FMT_MTD_CIMC.7: Extended TSF Private and Secret Key Export
- *FPT_WPF_EX.1: TSF Hardware Protection*
- *FPT_WPF_EX.2: TSF Disk Volume Protection*
- *FPT_WPF_EX.3: Removable USB Storage Device Protection*
- *FPT_WPF_EX.4: Network Access Protection*
- *FPT_WPF_EX.5: TPM Full Volume Encryption Support*
- *FPT_TRC_EXT.1: Internal TSF Data Consistency*
- *FPT_TST_EXT.1: TSF Testing*

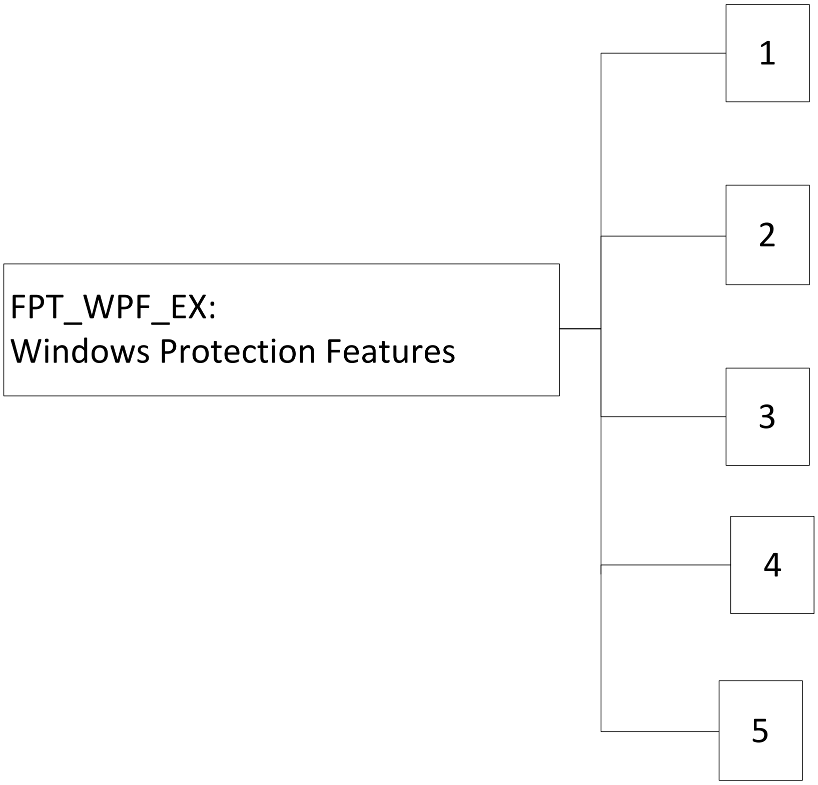
Of these, only the FPT_WPF_EX family of requirements is defined (below) in this Security Target. The others (italicized) are defined in the GPOSPP or (underlined) are defined in the (draft) *Certificate Issuing and Management Components For Basic Robustness Environments Protection Profile, Version 1.0, April 27, 2009* (CIMCPP) from which they have been borrowed.

5.1.1 Windows Protection Features (FPT_WPF_EX)

Family Behavior

The FPT_WPF_EX family of requirements is included within the Protection of the TSF (FPT) class as a collection of Windows-specific protection features that do not otherwise match well with pre-existing CC requirements.

Component Leveling



FPT_WPF_EX.1: TSF Hardware Protection provides the ability to utilize hardware features to prevent execution of memory that is not explicitly marked for the purpose of execution.

FPT_WPF_EX.2: TSF Disk Volume Protection provides the ability to protect (e.g., encrypt) the underlying system disk volume so that the TSF and user data remains protected even if the disk itself is compromised.

FPT_WPF_EX.3: Removable USB Storage Device Protection provides the ability to protect (e.g., encrypt) the content of removable USB storage devices and to require that such devices be protected in this regard before data can be written to them. Protection is ensured via credentials used to allow the content to be accessed.

FPT_WPF_EX.4: Network Access Protection provides the ability to restrict a given instance of the TOE (i.e., a client) from gaining unrestricted access to network resources when the TOE instance may be configured contrary to a defined compliance policy. When compliance is determined, network access credentials, such as IPsec certificates, VPN access, or DHCP configuration data will be provided to the client so that it can use them to access network resources.

FPT_WPF_EX.5: TPM Full Volume Encryption Support provides the ability for the TOE to make use of the capabilities of a supporting TPM chip in order to store encryption keys and to make them available only when appropriate conditions (i.e., system integrity) have been satisfied.

Management: FPT_WPF_EX.1, FPT_WPF_EX.2, FPT_WPF_EX.3, FPT_WPF_EX.4, and FTP_WPF_EX.5

Each of these functions can be configured per TSF data and as such TSF data configuration access should be constrained.

Audit: FPT_WPF_EX.1, FPT_WPF_EX.2, and FPT_WPF_EX.5

There are no auditable events foreseen.

Audit: FPT_WPF_EX.3 and FPT_WPF_EX.4

a) Minimal: Failed attempts to use the function.
b) Basic: Successful and failed attempts to use the function.

Each of these requirements is further defined (i.e., in terms of elements) in the next section. Note that none of these SFRs has any dependencies.

5.2 TOE Security Functional Requirements

This section specifies the SFRs for the TOE.

Table 5-1 TOE Security Functional Requirements

Requirement Class	Requirement Component
FAU: Security Audit	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User Identity Association
	FAU_SAR.1: Audit Review
	FAU_SAR.2: Restricted Audit Review
	FAU_SAR.3: Selectable Audit Review
	FAU_SEL.1: Selective Audit

Requirement Class	Requirement Component
	FAU_STG.1: Protected Audit Trail Storage
	FAU_STG.3: Action in Case of Possible Audit Data Loss
	FAU_STG.4: Prevention of Audit Data Loss
FCO: Communication	FCO_NRO_CIMC.3: Enforced Proof of Origin and Verification of Origin
	FCO_NRO_CIMC.4: Advanced Verification of Origin
FCS: Cryptographic Support	FCS_BCM_EXT.1: Baseline Cryptographic Module
	FCS_CKM.1a: Cryptographic Key Generation (for symmetric keys)
	FCS_CKM.1b: Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM.4: Cryptographic Key Destruction
	FCS_COA_EXT.1: Cryptographic Operations Availability
	FCS_COP.1a: Cryptographic Operation (for data encryption/decryption)
	FCS_COP.1b: Cryptographic Operation (for cryptographic signature)
	FCS_COP.1c: Cryptographic Operation (for cryptographic hashing)
	FCS_COP.1d: Cryptographic Operation (for data encryption/decryption)
	FCS_COP.1e: Cryptographic Operation (for cryptographic signature)

Requirement Class	Requirement Component
	FCS_COP.1f: Cryptographic Operation (ECDH key agreement)
	FCS_COP.1g: Cryptographic Operation (ECDSA key agreement)
	FCS_RBG_EXT.1: Random Number Generation
FDP: User Data Protection	FDP_ACC.2a: Complete Access Control
	FDP_ACC.2b: WEBUSER (WU) Complete Access Control
	FDP_ACC.2c: Content-Provider (CP) Complete Access Control
	FDP_ACC.2d: Mandatory Integrity Control Policy
	FDP_ACF.1a: Security Attribute Based Access Control
	FDP_ACF.1b: WEBUSER Access Control Functions
	FDP_ACF.1c: Content Provider Access Control Functions
	FDP_ACF.1d: Mandatory Integrity Control Functions
	FDP_ACF_CIMC.2: User Private Key Confidentiality Protection
	FDP_ACF_CIMC.3: User Secret Key Confidentiality Protection
	FDP_CIMC_CER.1: Certificate Generation
	FDP_CIMC_CRL.1: Certificate Revocation List Validation
	FDP_CIMC_CSE.1: Certificate Status Export

Requirement Class	Requirement Component
	FDP_CIMC_OCSP.1: OCSP Basic Response Validation
	FDP_ETC_CIMC.5: Extended User Private and Secret Key Export
	FDP_IFC.1a: IPSec Subset Information Flow Control
	FDP_IFC.1b: Windows Firewall Connection Subset Information Flow Control
	FDP_IFF.1a: IPSec Simple Security Attributes
	FDP_IFF.1b: Windows Firewall Connection Simple Security Attributes
	FDP_ITT.1: Basic Internal Transfer Protection
	FDP_RIP.2: Full Residual Information Protection
	FDP_UCT.1: WEBUSER Basic Data Exchange Confidentiality
	FDP_UIT.1: WEBUSER SFP Data Exchange Integrity
FIA: Identification and Authentication	FIA_AFL_EXT.1: Authentication Failure Handling
	FIA_ATD.1: User Attribute Definition
	FIA_SOS.1: Verification of Secrets
	FIA_UAU.1: Timing of Authentication
	FIA_UAU.6: Re-authenticating

Requirement Class	Requirement Component
	FIA_UAU.7: Protected Authentication Feedback
	FIA_UID.1: Timing of Identification
	FIA_USB.1: User-Subject Binding
FMT: Security Management	FMT_MOF.1a: Management of Security Functions Behavior (for specification of auditable events)
	FMT_MOF.1b: Management of Security Functions Behavior (for authentication data)
	FMT_MOF.1c: Management of Security Functions Behavior (for Certificate Services)
	FMT_MOF_CIMC.3: Extended Certificate Profile Management
	FMT_MOF_CIMC.5: Extended Certificate Revocation List Profile Management
	FMT_MOF_CIMC.6: OCSP Profile Management
	FMT_MSA.1a: Management of Security Attributes (for discretionary access control)
	FMT_MSA.1b: Management of Security Attributes (for object ownership)
	FMT_MSA.1c : Management of IPSec Object Security Attributes
	FMT_MSA.1d : Management of Windows Firewall Connection Object Security Attributes

Requirement Class	Requirement Component
	FMT_MSA.1e : Management of WEBUSER Object Security Attributes
	FMT_MSA.1f : Management of CONTENT-PROVIDER Object Security Attributes
	FMT_MSA.1g : Management of Mandatory Integrity Control Security Attributes
	FMT_MSA.2: Secure Security Attributes
	FMT_MSA.3a: Static Attribute Initialization
	FMT_MSA.3b: IPSec Static Attribute Initialization
	FMT_MSA.3c: Windows Firewall Connection Static Attribute Initialization
	FMT_MSA.3d: WEBUSER Static Attribute Initialization
	FMT_MSA.3e: CONTENT-PROVIDER Static Attribute Initialization
	FMT_MSA.3f: Mandatory Integrity Attribute Initialization
	FMT_MTD.1a: Management of TSF Data (for general TSF data)
	FMT_MTD.1b: Management of TSF Data (for audit data)
	FMT_MTD.1c: Management of TSF Data (for initialization of user security attributes)
	FMT_MTD.1d: Management of TSF Data (for modification of user security attributes, other than authentication data)

Requirement Class	Requirement Component
	FMT_MTD.1e: Management of TSF Data (for modification of authentication data)
	FMT_MTD.1f: Management of TSF Data (for reading of authentication data)
	FMT_MTD.1g: Management of TSF Data (for critical cryptographic security parameters)
	FMT_MTD_CIMC.4: TSF Private Key Confidentiality Protection
	FMT_MTD_CIMC.5: TSF Secret Key Confidentiality Protection
	FMT_MTD_CIMC.7: Extended TSF Private and Secret Key Export
	FMT_REV.1a: Revocation (to authorized administrators)
	FMT_REV.1b: Revocation (to owners and authorized administrators)
	FMT_SAE.1: Time-Limited Authorization
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.1: Security Roles
FPT: Protection of the TSF	FPT_ITT.1: Basic Internal TSF Data Transfer Protection
	FPT_ITT.3: TSF Data Integrity Monitoring
	FPT_RCV.1: Manual Recovery

Requirement Class	Requirement Component
	FPT_WPF_EX.1: TSF Hardware Protection
	FPT_WPF_EX.2: TSF Disk Volume Protection
	FPT_WPF_EX.3: Removable USB Storage Device Protection
	FPT_WPF_EX.4: Network Access Protection
	FPT_WPF_EX.5: TPM Full Volume Encryption Support
	FPT_STM.1: Reliable Time Stamps
	FPT_TRC_EXT.1: Internal TSF Data Consistency
	FPT_TST_EXT.1: TSF Testing
FRU: Resource Utilization	FRU_RSA.1: Maximum Quotas
FTA: TOE Access	FTA_MCS.1: Basic Limitation on Multiple Concurrent Sessions
	FTA_SSL.1: TSF-initiated Session Locking
	FTA_SSL.2: User-initiated Locking
	FTA_SSL.3: WEBUSER TSF-Initiated Termination
	FTA_TAB.1: Default TOE Access Banners
	FTA_TAH.1: TOE Access History
	FTA_TSE.1: TOE Session Establishment

Requirement Class	Requirement Component
FTP: Trusted Path/Channels	FTP_TRP.1: Trusted Path

5.2.1 Security Audit (FAU)

5.2.1.1 Audit Data Generation (FAU_GEN.1)

FAU_GEN.1.1 The TSF shall be able to generate an audit record of the following auditable events:

a) Start-up and shutdown of the audit functions,
b) Start-up and shutdown of the TOE,
c) Uses of special permissions that circumvent the access control policies,
d) All auditable events listed in Table 5-2, and
e) All auditable events for the minimal level of audit.

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, the additional information in Table 5-2.

Table 5-2 Audit Events and Information

Requirement	Audit events prompted by requirement	Additional Information in audit record
Audit Data Generation (FAU_GEN.1)	(none)	(none)
User Identity Association (FAU_GEN.2)	(none)	(none)
Audit Review (FAU_SAR.1)	• Opening the audit records.	Name of object (audit log file)
Restricted Audit Review (FAU_SAR.2)	• Unsuccessful attempts to read information from the audit records.	(none)
Selectable Audit Review (FAU_SAR.3)	(none)	(none)
Selective Audit (FAU_SEL.1)	• All modifications to the audit configuration that occur while the audit collection functions are operating.	(none)
Protected Audit Trail Storage (FAU_STG.1)	(none)	(none)
Action in Case of Possible	• Actions taken due to exceeding	Message sent to administrator

Requirement	Audit events prompted by requirement	Additional Information in audit record
Audit Data Loss (FAU_STG.3)	of a threshold.	
Prevention of Audit Data Loss (FAU_STG.4)	(none)	(none)
Enforced Proof of Origin and Verification of Origin (FCO_NRO_CIMC.3)	(none)	(none)
Advanced Verification of Origin (FCO_NRO_CIMC.4)	(none)	(none)
Baseline Cryptographic Module (FCS_BCM_EXT.1)	• Failure of the cryptographic operation.	(none)
Cryptographic Key Generation (FCS_CKM.1)	• Success and failure of the key generation process **for asymmetric keys**.	Identity of the subject.
Cryptographic Key Destruction (FCS_CKM.4)	• Failure of key zeroization process.	Identity of subject requesting or causing zeroization, identity of object or entity being cleared.
Cryptographic Operations Availability (FCS_COA_EXT.1)	(none)	(none)
Cryptographic Operation (for data encryption/decryption) (FCS_COP.1a)	(none)	(none)
Cryptographic Operation (for cryptographic signature) (FCS_COP.1b)	• Failure in cryptographic signature.	Identity of the subject.
Cryptographic Operation (for cryptographic hashing) (FCS_COP.1c)	(none)	Type and cryptographic mode of operation, name of object being hashed.
Cryptographic Operation (for data encryption/decryption) (FCS_COP.1d)	(none)	(none)
Cryptographic Operation (for cryptographic signature) (FCS_COP.1e)	• Failure of the operation.	(none)
Cryptographic Operation (ECDH Key Agreement) (FCS_COP.1f)	• Failure of the operation.	.(none)
Cryptographic Operation (ECDSA Key Agreement) (FCS_COP.1g)	• Failure of the operation.	Identity of the subject.
Random Number Generation (FCS_RBG_EXT.1)	• Failure in the randomization process.	(none)
Complete Access Control (FDP_ACC.2a)	(none)	(none)

Requirement	Audit events prompted by requirement	Additional Information in audit record
WEBUSER (WU) Complete Access Control (FDP_ACC.2b)	(none)	(none)
Content-Provider (CP) Complete Access Control (FDP_ACC.2c)	(none)	(none)
Mandatory Integrity Control Policy (FDP_ACC.2d)	(none)	(none)
Security Attribute Based Access Control (FDP_ACF.1a)	• All requests to perform an operation on an object covered by the SFP. • Use of privilege to bypass the access control mechanism.	The name of the object being accessed.
WEBUSER Access Control Functions (FDP_ACF.1b)	• Successful and unsuccessful requests to perform an operation on an object covered by the SFP.	(none)
Content Provider Access Control Functions (FDP_ACF.1c)	• Successful and unsuccessful requests to perform an operation on an object covered by the SFP.	(none)
Mandatory Integrity Control Functions (FDP_ACF.1d)	• All requests to perform an operation on an object covered by the SFP.	(none)
User Private Key Confidentiality Protection (FDP_ACF_CIMC.2)	(none)	(none)
User Secret Key Confidentiality Protection (FDP_ACF_CIMC.3)	(none)	(none)
Certificate Generation (FDP_CIMC_CER.1)	• Approval and issuance of a new certificate. • Rejection of a request for a new certificate.	Information on requestor, approver, and if approved certificate information.
Certificate Revocation List Validation (FDP_CIMC_CRL.1)	• Publication of a new certificate revocation list	The base CRL, CRL number, key number, next publication date, publication URL.
Certificate Status Export (FDP_CIMC_CSE.1)	(none)	(none)
OCSP Basic Response Validation (FDP_CIMC_OCSP.1)	•Arrival of a request for the OCSP Responder.	(none)
Extended User Private and Secret Key Export (FDP_ETC_CIMC.5)	(none)	(none)
IPSec Subset Information	(none)	(none)

Requirement	Audit events prompted by requirement	Additional Information in audit record
Flow Control (FDP_IFC.1a)		
Windows Firewall Connection Subset Information Flow Control (FDP_IFC.1b)	(none)	(none)
IPSec Simple Security Attributes (FDP_IFF.1a)	• Decisions to permit requested information flows. [19]	(none)
Windows Firewall Connection Simple Security Attributes (FDP_IFF.1b)	• Decisions to permit requested information flows.[20]	(none)
Basic Internal Transfer Protection (FDP_ITT.1)	• Successful transfers of user data.	Identification of the protection method used.
Full Residual Information Protection (FDP_RIP.2)	(none)	(none)
WEBUSER Basic Data Exchange Confidentiality (FDP_UCT.1)	• Use of the data exchange mechanisms.	The identity of any user or subject.
WEBUSER SFP Data Exchange Integrity (FDP_UIT.1)	• Use of the data exchange mechanisms.	The identity of any user or subject.
Authentication Failure Handling (FIA_AFL_EXT.1)	• The reaching of the threshold for the unsuccessful authentication attempts. • The action taken (disable for non-administrators, delay for administrator). • The re-enablement of disabled non-administrative accounts.	(none)
User Attribute Definition (FIA_ATD.1)	(none)	(none)
Verification of Secrets (FIA_SOS.1)	• Rejection by the TSF of any tested secret.	(none)
Timing of Authentication (FIA_UAU.1)	• All use of the authentication mechanism.	Origin of the attempt (e.g., terminal identifier, source IP address)
Re-authenticating (FIA_UAU.6)	• All re-authentication attempts when changing authentication data	Origin of the attempt (e.g., terminal identifier, source IP address)
Protected Authentication Feedback (FIA_UAU.7)	(none)	(none)
Timing of Identification (FIA_UID.1)	• All use of the user identification mechanism	Provided user identity, origin of the attempt (e.g., terminal

[19] The result of the decision is either a successful or unsuccessful IPSec negotiation.
[20] The result of the decision is either a successful or unsuccessful IPSec negotiation.

Requirement	Audit events prompted by requirement	Additional Information in audit record
		identifier, source IP address)
User-Subject Binding (FIA_USB.1)	• Binding of user security attributes to a subject (e.g. creation of a subject).	(none)
Management of Security Functions Behavior (for specification of auditable events) (FMT_MOF.1a)	• All modifications in the behavior of the functions in the TSF.	The old and new values for audit events specified by this function.
Management of Security Functions Behavior (for authentication data) (FMT_MOF.1b)	• All modifications in the behavior of the functions in the TSF.	(none)
Extended Certificate Profile Management (FMT_MOF_CIMC.3)	• All changes to the certificate profile.	The changes to the certificate profile.
Extended Certificate Revocation List Profile Management (FMT_MOF_CIMC.5)	(none)	(none)
OCSP Profile Management (FMT_MOF_CIMC.6)	• All changes to the OCSP profile.	The changes to the OCSP profile.
Management of Security Attributes (FMT_MSA.1a)	• All modifications of the values of security attributes.	The name of the object, the old and new values of the attributes
Management of Security Attributes (FMT_MSA.1b)	• All modifications of the values of security attributes.	The name of the object, the old and new values of the attributes
Management of IPSec Object Security Attributes (FMT_MSA.1c)	(none)	(none)
Management of Windows Firewall Connection Object Security Attributes (FMT_MSA.1d)	(none)	(none)
Management of WEBUSER Object Security Attributes (FMT_MSA.1e)	(none)	(none)
Management of CONTENT-PROVIDER Object Security Attributes (FMT_MSA.1f)	(none)	(none)
Management of Mandatory Integrity Control Security Attributes (FMT_MSA.1g)	(none)	(none)
Secure Security Attributes (FMT_MSA.2)	• All modifications of the values of security attributes.	All offered and rejected values for a security attribute.

Requirement	Audit events prompted by requirement	Additional Information in audit record
Static Attribute Initialization (FMT_MSA.3a)	• Modifications of the default setting of permissive or restrictive rules. • All modifications of the initial values of security attributes.	The old and new values of the attributes.
IPSec Static Attribute Initialization (FMT_MSA.3b)	(none)	(none)
Windows Firewall Connection Static Attribute Initialization (FMT_MSA.3c)	(none)	(none)
WEBUSER Static Attribute Initialization (FMT_MSA.3d)	(none)	(none)
CONTENT-PROVIDER Static Attribute Initialization (FMT_MSA.3e)	(none)	(none)
Mandatory Integrity Attribute Initialization (FMT_MSA.3f)	(none)	(none)
Management of TSF Data (for general TSF data) (FMT_MTD.1a)	• All modifications of the values of TSF data.	The old and new values of the TSF data.
Management of TSF Data (for audit data) (FMT_MTD.1b)	• Actions taken with respect to the audit records.	The specific action that was performed.
Management of TSF Data (for initialization of user security attributes) (FMT_MTD.1c)	• All initializations of the values of user security attributes.	The initial values for the user security attributes.
Management of TSF Data (for modification of user security attributes, other than authentication data) (FMT_MTD.1d)	• All modifications of the values of user security attributes.	The old and new values of the attributes.
Management of TSF Data (for modification of authentication data) (FMT_MTD.1e)	• All actions associated with modifications of the values of authentication data.	(none)
Management of TSF Data (for reading of authentication data) (FMT_MTD.1f)	(none)	(none)
Management of TSF Data (for critical cryptographic security parameters) (FMT_MTD.1g)	• All actions associated with modifications of the values of critical cryptographic security parameters.	The old and new values of the parameters, excluding any sensitive information, such as secret or private keys.

Requirement	Audit events prompted by requirement	Additional Information in audit record
TSF Private Key Confidentiality Protection (FMT_MTD_CIMC.4)	(none)	(none)
TSF Secret Key Confidentiality Protection (FMT_MTD_CIMC.5)	(none)	(none)
Extended TSF Private and Secret Key Export (FMT_MTD_CIMC.7)	(none)	(none)
Revocation (to authorized administrators) (FMT_REV.1a)	• All attempts to revoke security attributes.	The security attributes that are attempting to be revoked
Revocation (to owners and authorized administrators) (FMT_REV.1b)	• All attempts to revoke security attributes.	The security attributes that are attempting to be revoked, the object with which the security attributes are associated.
Time-Limited Authorization (FMT_SAE.1)	• Specification of the expiration time for an attribute. • Action taken due to attribute expiration.	(none)
Specification of Management Functions (FMT_SMF.1)	• Use of the management functions.	(none)
Security Roles (FMT_SMR.1)	• Modifications to the group of users that are part of a role.	The role the user is associated with or disassociated from.
Basic Internal TSF Data Transfer Protection (FPT_ITT.1)	(none)	(none)
TSF Data Integrity Monitoring (FPT_ITT.3)	• Detection of modification of TSF data.	Network address of source and destination of the transfer.
Manual Recovery (FPT_RCV.1)	• The fact that a failure or service discontinuity occurred. • Resumption of the regular operation.	Type of failure or service discontinuity
TSF Hardware Protection (FPT_WPF_EX.1)	(none)	(none)
TSF Disk Volume Protection (FPT_WPF_EX.2)	(none)	(none)
Removable USB Storage Device Protection (FPT_WPF_EX.3)	•Failed attempts to access device content.	(none)
Network Access Protection (FPT_WPF_EX.4)	•Failed attempts to access a network due to policy non-conformance.	(none)

Requirement	Audit events prompted by requirement	Additional Information in audit record
TPM Full Volume Encryption Support (FPT_WPF_EX.5)	(none)	(none)
Reliable Time Stamps (FPT_STM.1)	• Setting the time to a specific value.	The old and new values for the time.
Internal TSF Data Consistency (FPT_TRC_EXT.1)	(none)	(none)
TSF Testing (for cryptography) (FPT_TST.1)	• Execution of the cryptography self-tests.	For each test, the identification of the test and the results of that test.
Maximum Quotas (FRU_RSA.1)	• Rejection of allocation operation due to persistent storage limits.	Object or other entity associated with failed allocation operation.
Basic limitation on multiple concurrent sessions (FTA_MCS.1)	• Rejection of a new session based on the limitation of multiple concurrent sessions. • Setting the limit on the number of multiple concurrent sessions by an authorized administrator.	The old and new values of the number of multiple concurrent sessions (for setting the session limit).
TSF-Initiated Session Locking (FTA_SSL.1)	• Locking of an interactive session by the session locking mechanism. • Any attempts at unlocking of an interactive session.	(none)
User-Initiated Locking (FTA_SSL.2)	• Locking of an interactive session by the session locking mechanism. • Any attempts at unlocking of an interactive session.	(none)
WEBUSER TSF-Initiated Termination (FTA_SSL.3)	• Termination of an interactive session by the session locking mechanism.	(none)
Default TOE Access Banners (FTA_TAB.1)	(none)	(none)
TOE Access History (FTA_TAH.1)	(none)	(none)
TOE Session Establishment (FTA_TSE.1)	• Denial of a session establishment due to the session establishment mechanism.	(none)
Trusted Path (FTP_TRP.1)	• Failures of the trusted path functions.	Identification of the user associated with all trusted path failures, if available.

5.2.1.2 User Identity Association (FAU_GEN.2)

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.2.1.3 Audit Review (FAU_SAR.1)

FAU_SAR.1.1 The TSF shall provide authorized administrators with the capability to read all audit information from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the authorized administrator to interpret the information using a tool to access the audit records.

5.2.1.4 Restricted Audit Review (FAU_SAR.2)

FAU_SAR.2.1 The TSF shall prohibit all users read access to the audit records, except those users that have been granted explicit read-access.

5.2.1.5 Selectable Audit Review (FAU_SAR.3)

FAU_SAR.3.1 The TSF shall provide the ability to perform searches **and sorting** of audit data based on the following attributes:

a) user identity,
b) object identity **(searches only)**,
c) date of the event,
d) time of the event,
e) type of event,
f) success of auditable security events, and
g) failure of auditable security events.

5.2.1.6 Selective Audit (FAU_SEL.1)

FAU_SEL.1.1 The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a) object identity,
b) user identity,
c) host identity,
d) event type,
e) success of auditable security events, and
f) failure of auditable security events.

5.2.1.7 Protected Audit Trail Storage (FAU_STG.1)

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to prevent modifications to the stored audit records in the audit trail.

5.2.1.8 Action in Case of Possible Audit Data Loss (FAU_STG.3)

FAU_STG.3.1 The TSF shall notify an authorized administrator of the possible audit data loss if the audit trail exceeds an authorized administrator selectable, pre-defined limit.

5.2.1.9 Prevention of Audit Data Loss (FAU_STG.4)

FAU_STG.4.1 The TSF shall ***[prevent audited events, except those taken by the authorized user with special rights[21]]*** and **[generate an alarm to the authorized administrator]** if the audit trail is full.

5.2.2 Communication (FCO)

5.2.2.1 Enforced Proof of Origin and Verification of Origin (FCO_NRO_CIMC.3)

FCO_NRO_CIMC.3.1 The TSF shall enforce the generation of evidence of origin for certificate status information ~~and all other security-relevant information~~[22] at all times.

FCO_NRO_CIMC.3.2 The TSF shall be able to relate the identity and [**no other attributes**] of the originator of the information, and the security-relevant portions of the information to which the evidence applies.

FCO_NRO_CIMC.3.3 The TSF shall verify the evidence of origin of information for all ~~security-relevant~~ **certificate status** information.

5.2.2.2 Advanced Verification of Origin (FCO_NRO_CIMC.4)

FCO_NRO_CIMC.4.1 The TSF shall, for initial certificate registration messages sent by the certificate subject, only accept messages protected using an authentication code, keyed hash, or digital signature algorithm.

FCO_NRO_CIMC.4.2 The TSF shall, for all other security-relevant information, only accept the information if it was signed using a digital signature algorithm.

5.2.3 Cryptographic support (FCS)

5.2.3.1 Baseline Cryptographic Module (FCS_BCM_EXT.1)

FCS_BCM_EXT.1.1 All FIPS-approved cryptographic functions implemented by the TOE shall be implemented in a cryptomodule that is FIPS 140-2 validated, and perform the specified cryptographic functions in a FIPS-approved mode of operation. The FIPS 140-2 validation shall include an algorithm validation certificate for all FIPS-approved cryptographic functions.

5.2.3.2 Cryptographic Key Generation (for symmetric keys) (FCS_CKM.1a)

FCS_CKM.1a.1 The TSF shall generate cryptographic keys using a FIPS-Approved Random Number Generator as specified in FCS_RBG_EXT.1, and provide integrity protection to generated keys that leave the cryptomodule in accordance with NIST SP 800-57 'Recommendation for Key Management—Part 1:

[21] In this case the "authorized user with special rights" is the authorized administrator.

[22] While the source PP indicates proof of origin applies to all security-relevant information, that is too broad for other product types and that caveat has been removed and subsequent elements modified accordingly..

General,' paragraph 6.2.2.2a. in the following manner**: [cryptographic signature and cryptographic hashing services]**.

5.2.3.3 Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1b)

FCS_CKM.1b.1 The TSF shall generate asymmetric cryptographic keys in accordance with domain parameter sizes ***[for rDSA-based keys, [at least 2048 bits], for ECDSA-based keys, [256 bits, 384 bits, 512 bits]]***, **for ECDH between 384 and 4096 bits, and for DSA at least 1024 bits** that meet the following: FIPS 140-2.

5.2.3.4 Cryptographic Key Destruction (FCS_CKM.4)

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a cryptographic key zeroization method that meets the following: Key zeroization requirements of FIPS PUB 140-2, 'Security Requirements for Cryptographic Modules'.

5.2.3.5 Cryptographic Operations Availability (FCS_COA_EXT.1)

FCS_COA_EXT.1.1 The TSF shall provide the following cryptographic operations to applications:

a) Encryption/Decryption,
b) Cryptographic Signature (Digital Signature),
c) Hashing, and
d) [Random Number Generation and Key Agreement].

5.2.3.6 Cryptographic Operation (for data encryption/decryption) (FCS_COP.1a)

FCS_COP.1a.1 The TSF shall perform encryption and decryption using the FIPS-approved security function AES algorithm operating in **[ECB, CBC, CFB8, CCM, and GCM modes]** and cryptographic key size of ***[128 bits, 192 bits, 256 bits]*** that meets FIPS 140-2.

5.2.3.7 Cryptographic Operation (for cryptographic signature) (FCS_COP.1b)

FCS_COP.1b.1 The TSF shall perform cryptographic signature services using the FIPS-approved security function ***[RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of [at least 2048 bits], or Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of [256 bits, 384 bits, or 521 bits], using only the NIST curve(s) [P-256, P-384, P-521 as defined in FIPS PUB 186-3, 'Digital Signature Standard']]*** that meets FIPS 140-2.

5.2.3.8 Cryptographic Operation (for cryptographic hashing) (FCS_COP.1c)

FCS_COP.1c.1 The TSF shall perform cryptographic hashing services in accordance with ***[SHA 256, SHA 384, SHA 512]*** and message digest sizes ***[256, 384, or 512]*** bits that meet the following: FIPS 140-2.

5.2.3.9 Cryptographic Operation (for data encryption/decryption) (FCS_COP.1d)

FCS_COP.1d.1 The TSF shall perform **[encryption and decryption]** in accordance with a specified cryptographic algorithm **[Triple DES (3DES) ECB, CBC, and CFB modes]** and cryptographic key sizes **[168-bits]** that meet the following: **[FIPS 46-3]**.[23]

5.2.3.10 Cryptographic Operation (for cryptographic signature) (FCS_COP.1e)

FCS_COP.1e.1 The TSF shall perform **[digital signing]** in accordance with a specified cryptographic algorithm **[Digital Signature Algorithm (DSA)]** and cryptographic key size **[at least 1024 bits]** that meet the following: **[FIPS 186-2]**.[24]

5.2.3.11 Cryptographic Operation (ECDH key agreement) (FCS_COP.1f)

FCS_COP.1f.1 The TSF shall perform **[key agreement]** in accordance with a specified cryptographic algorithm **[Elliptic Curve Diffie Hellman (ECDH) key agreement protocol]** and cryptographic key sizes [between 384 and 4096 bits] that meet the following: **[NIST SP 800-56A and FIPS 140-2]**.

5.2.3.12 Cryptographic Operation (ECDSA key agreement) (FCS_COP.1g)

FCS_COP.1g.1 The TSF shall perform **[key agreement]** in accordance with a specified cryptographic algorithm **[Elliptic Curve Digital Signature Algorithm for key agreement with NIST P curves: P-256, P-384, and P-521 (ECDSA)]** and cryptographic key **sizes [256, 384, and 521, respectively]** that meet the following: **[ANSI X9.62]**.[25]

5.2.3.13 Random Number Generation (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with ***[NIST Special Publication 800-90]*** implemented in a FIPS-validated cryptomodule operating in FIPS mode seeded by an entropy source that accumulates entropy from ***[a combination of hardware-based and software-based noise sources]***.

Application Note: When a Trusted Platform Module (TPM) chip is available in the underlying hardware, the TOE will utilize its hardware-based noise source features to improve random number generation.

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of ***[256 bits]*** of entropy at least equal to the greatest bit length of the keys that it will generate.

5.2.4 User Data Protection (FDP)

5.2.4.1 Complete Access Control (FDP_ACC.2a)

FDP_ACC.2a.1 The TSF shall enforce the Discretionary Access Control policy on all subjects and all named objects and all operations among them.

[23] Note that these operations are performed within a FIPS 140-evaluated cryptographic module, See FIPS 140-2 certificates 1329 and 1336.

[24] Note that these operations are performed within a FIPS 140-evaluated cryptographic module, See FIPS 140-2 certificates 1329 and 1336.

[25] Note that these operations are performed within a FIPS 140-evaluated cryptographic module, See FIPS 140-2 certificates 1329 and 1336.

FDP_ACC.2a.2 The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

5.2.4.2 WEBUSER (WU) Complete Access Control (FDP_ACC.2b)

FDP_ACC.2b.1 The TSF shall enforce the **[WEBUSER SFP]** on **[**

a) **Web Server subjects: web users – processes acting on behalf of users (which are users of the OS part of the TOE/TSF) requesting web access and**
b) **Web Server objects: web server content (served by the Web Server part of TSF over http:// or https://)]**

and all operations among subjects and objects covered by the SFP.

FDP_ACC.2b.2 The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

5.2.4.3 Content-Provider (CP) Complete Access Control (FDP_ACC.2c)

FDP_ACC.2c.1 The TSF shall enforce the **[CONTENT-PROVIDER (CP) SFP]** on **[**

a) **Content Provider subjects: Content-Providers - processes acting on behalf of users (which are users of the OS part of the TOE/TSF) and**
b) **Content Provider objects: Web Server Content (served by the Web Server part of TSF over http:// or https://)]**

and all operations among subjects and objects covered by the SFP.

FDP_ACC.2c.2 The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

5.2.4.4 Mandatory Integrity Control Policy (FDP_ACC.2d)

FDP_ACC.2d.1 The TSF shall enforce the **[Mandatory Integrity Control Policy] on [**

a) **subjects: processes acting on the behalf of users and**
b) **objects: Event, Keyed Event, Event Pair, I/O Completion Port, Job, Key, Mutant, Mailslot, Named Pipe, NTFS Directory, NTFS File, Object Directory, Process, Section, Semaphore, Symbolic Link, Thread, Timer, and Tokens]**

and all operations among them subjects and objects covered by the SFP.

FDP_ACC.2d.2 The TSF shall ensure that all operations between any subject controlled by the TSF and any object controlled by the TSF are covered by an access control SFP.

5.2.4.5 Security Attribute Based Access Control (FDP_ACF.1a)

FDP_ACF.1a.1 The TSF shall enforce the Discretionary Access Control policy to named objects based on the following types of subject and object security attributes:

a) the authorized user identity and group membership(s) associated with a subject;

b) the [authorized user (or group) identity, access operations] pairs associated with a named object; and

c) ***[no other attributes]***.

FDP_ACF.1a.2 The TSF shall enforce the following rules to determine if an operation among subjects and named objects is allowed:

The Discretionary Access Control policy mechanism shall, either by explicit authorized user action or by default, provide that named objects are protected from unauthorized access according to the following ordered rules:

1) If the requested mode of access is denied to that authorized user, deny access.
2) If the requested mode of access is permitted to that authorized user, permit access.
3) If the requested mode of access is denied to every group of which the authorized user is a member, deny access
4) If the requested mode of access is permitted to any group of which the authorized user is a member, grant access
5) Else deny access.

FDP_ACF.1a.3 The TSF shall explicitly authorize access of subjects to named objects based on the following additional rules:

a) Authorized administrators must follow the above-stated Discretionary Access Control policy, except after taking the following specific actions: **[**
   - **Request to change the owner of an object,**
   - **Request to backup a file or registry key on the local system,**
   - **Request to restore a file or registry key onto the local system,**
   - **Request to synchronize Active Directory objects to another domain controller]**,

b) The enforcement mechanism (e.g., access control lists) shall allow authorized users to specify and control sharing of named objects by individual user identities and group identities, and

c) [If an object has no access control list the object is not protected and any requested access is granted, and

d) For encrypted file objects, in addition to meeting FDP_ACF.1.2, the user must have a private key that can decrypt the FEK[26] associated with the file.].

FDP_ACF.1a.4 The TSF shall explicitly deny access of subjects to named objects based on the following rules: **[If an object has an assigned, but empty access control list no access is granted]**.

5.2.4.6 WEBUSER Access Control Functions (FDP_ACF.1b)

FDP_ACF.1b.1 The TSF shall enforce the **[WEBUSER SFP]** to objects based on the following: **[**

a) Web Server Subjects – web users – process on behalf of users (which are users of the OS part of the TOE/TSF) requesting access:

[26] File Encrypting Key

   a. **the user identity and**
   b. **group membership(s) associated with the subject and**

b) **Web Server Objects – web server content (served by the Web Server part of the TSF over http:// or https://):**
   a. **the DACL associated with the object,**
   b. **the web permissions associated with an object, and**
   c. **the URL authorization associated with an object].**

FDP_ACF.1b.2 The TSF shall enforce the following **ordered** rules to determine if an operation among controlled subjects and controlled objects is allowed: **[**

a) **For (Web Server) controlled-access content:**
   a. **If the requested access is denied by the file's DACL associated with the web content to that web user, deny access.**
   b. **If the requested access is something other than read access, deny access.**
   c. **If read-only access is permitted to that authorized web user by the file's DACL associated with the web content, grant access.**
   d. **Otherwise, deny access.**

b) **For (Web Server) public content:**
   a. **If the requested access is something other than read access, deny access.**
   b. **Grant read-only access to web user].**

FDP_ACF.1b.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[**

a) **A web user trying to access an object URL must be authorized to the operation "access URL", if URL authorization is configured for the object.**
b) **A web user may read web server content if the web permission associated with the object allows read access.**
c) **A web user may change web server content if the web permission associated with the object allows write access.**
d) **A web user may access the source of a web server content if the web permission associated with the object allows access to the source.**
e) **A web user may view web server content file lists and collections if the web permission associated with the object allows browsing access].**

FDP_ACF.1b.4 The TSF shall explicitly deny access of subjects to objects based on the **following rules: [**

a) **If a web user uses http:// instead of https:// and the web permission associated with the object requires TLS/SSL.**
b) **If a web user does not use a client certificate and the web permission associated with the object requires TLS/SSL and a certificate.**
c) **If the web user's certificate is revoked or is invalid and the web permission associated with the object requires TLS/SSL and a negotiated certificate or requires a certificate.**

- **d) If the authorization setting of a web user determined by an authentication provider does not match the configured authorization setting associated with the object.**
- **e) If the client certificate mapping setting of a web user determined by an authentication provider does not must match the configured client certificate mapping setting associated with the object.**
- **f) If the web permission requested is not supported (other than those permissions identified in FDP_ACF.1b.3)].**

Application Note: “Public content” is web content that can be accessed without authentication.

Application Note: The WEBUSER Access Control function describes how a HTTP(S) client retrieves content from a web server. The FDP_ACF.1b.2 functional requirement describes the relationship between web access and the underlying DAC policy of the NTFS-based files that represent the web server’s content. The FDP_ACF.1b.3 function describes additional access control authorizations that occur before the DAC access check.

5.2.4.7 Content Provider Access Control Functions (FDP_ACF.1c)

FDP_ACF.1c.1 The TSF shall enforce the **[CONTENT-PROVIDER SFP]** to objects based on the following: **[**

- **a) Content Providers – processes acting on behalf of users (which are users of the OS part of the TOE/TSF) (which are just users of the OS part of the TOE/TSF):**
  - **a. the user identity and**
  - **b. group membership(s) associated with a subject and**
- **b) Web Server Content (served by the Web Server part of the TSF over http:// or https://):**
  - **a. the web permissions associated with an object,**
  - **b. the DACL associated with the object, and**
  - **c. the URL authorization associated with an object].**

FDP_ACF.1c.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: **[**

- **a) The Web Server part of the TOE shall restrict the ability to create or modify content to only those content providers authorized by an authorized administrator.**
- **b) For (Web Server) controlled-access content:**
  - **a. If the requested access is denied by the file’s DACL associated with the web content to that web user, deny access.**
  - **b. If the requested access is something other than read access, deny access.**
  - **c. If read-only access is permitted to that authorized web user by the file’s DACL associated with the web content, grant access**
  - **d. Otherwise, deny access.**
- **c) For (Web Server) public content:**
  - **a. If the requested access is something other than read access, deny access.**
  - **b. Grant read-only access to web user].**

FDP_ACF.1c.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: **[**

a) **A content provider trying to access an object URL must be authorized to the operation "access URL" if the URL Authorization is configured for the object.**
b) **A content provider may read web server content if the web permission associated with the object allows read access.**
c) **A content provider may change web server content if the web permission associated with the object allows write access.**
d) **A content provider may execute web server content if the web permission associated with the object allows execute access.**
e) **A content provider may access the source of web server content if the web permission associated with the object allows access to the source.**
f) **A content provider may view web server content file lists and collections if the web permission associated with the object allows browsing access].**

FDP_ACF.1c.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[**

a) **If a content provider uses http:// instead of https:// and the web permission associated with the object requires TLS/SSL.**
b) **If a content provider does not use a client certificate and the web permission associated with the object requires TLS/SSL and a certificate.**
c) **If the content provider's certificate is revoked or is invalid and the web permission associated with the object requires SSL and that a certificate be negotiated, or requires TLS/SSL and a certificate.**
d) **If the authorization setting of a content provider determined by an authentication provider does not match the configured authorization setting associated with the object.**
e) **If the client certificate mapping setting of a content provider determined by an authentication provider does not must match the configured client certificate mapping setting associated with the object.**
f) **If the web permission requested is not supported (other than those permissions identified in FDP_ACF.1c.3)].**

Application Note: "Public content" is web content that can be accessed without authentication

Application Note: The Content Provider Access Control function describes how a HTTP(S) client uploads or modifies content on a web server. The FDP_ACF.1c.2 functional requirement describes the relationship between web access and the underlying DAC policy of the NTFS-based files that represent the web server's content. The FDP_ACF.1c.3 function describes additional access control authorizations that occur before the DAC access check.

5.2.4.8 Mandatory Integrity Control Functions (FDP_ACF.1d)

FDP_ACF.1d.1 The TSF shall enforce the **[Mandatory Integrity Control Policy]** to objects based on the following: **[**

a) **The integrity label and mandatory policy associated with a subject and**
b) **The integrity label and mandatory policy associated with an object]**.

FDP_ACF.1d.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed**: [**

a) **Write access is allowed if the subject integrity label is greater than or equal to the object integrity label OR the object mandatory policy does not indicate "SYSTEM_MANDATORY_LABEL_NO_WRITE_UP".**
b) **Read access is allowed if the subject integrity label is greater than or equal to the object integrity label OR the object mandatory policy does not indicate "SYSTEM_MANDATORY_LABEL_NO_READ_UP".**
c) **Execute access is allowed if the subject integrity label is greater than or equal to the object integrity label OR the object mandatory policy does not indicate "SYSTEM_MANDATORY_LABEL_NO_EXECUTE_UP"]**.

FDP_ACF.1d.3 The TSF shall explicitly authorize access of subjects to objects based in the following additional rules**: [The mandatory policy associated with the subject does not indicate "TOKEN_MANDATORY_POLICY_NO_WRITE_UP"]**.

FDP_ACF.1d.4 The TSF shall explicitly deny access of subjects to objects based on the following additional rules: **[no explicit denial rules]**.

5.2.4.9 User Private Key Confidentiality Protection (FDP_ACF_CIMC.2)

FDP_ACF_CIMC.2.1 **User** ~~CIMS personnel~~ private keys shall be stored in a FIPS 140-2 validated cryptographic module or stored in encrypted form. If **User** ~~CIMS personnel~~ private keys are stored in encrypted form, the encryption shall be performed by the FIPS 140-2 validated cryptographic module.

FDP_ACF_CIMC.2.2 If certificate subject private keys are stored in the TOE, they shall be encrypted using a Long Term Private Key Protection Key. The encryption shall be performed by the FIPS 140-2 validated cryptographic module.

5.2.4.10 User Secret Key Confidentiality Protection (FDP_ACF_CIMC.3)

FDP_ACF_CIMC.3.1 User secret keys stored within the **TOE** ~~CIMC~~, but not within a FIPS 140-2 validated cryptographic module, shall be stored in encrypted form. The encryption shall be performed by the FIPS 140-2 validated cryptographic module.

5.2.4.11 Certificate Generation (FDP_CIMC_CER.1)

FDP_CIMC_CER.1.1 The TSF shall only generate certificates whose format complies with [**the X.509 standard for public key certificates**].

FDP_CIMC_CER.1.2 The TSF shall only generate certificates that are consistent with the currently defined certificate profile.

FDP_CIMC_CER.1.3 The TSF shall verify that the prospective certificate subject possesses the private key that corresponds to the public key in the certificate request before issuing a certificate, unless the public/private key pair was generated by the TSF, whenever the private key may be used to generate digital signatures.

FDP_CIMC_CER.1.4 If the TSF generates X.509 public key certificates, it shall only generate certificates that comply with requirements for certificates as specified in ITU-T Recommendation X.509. At a minimum, the TSF shall ensure that:

a) The **version** field shall contain the integer **0**, **1**, **2, or 3**.

b) If the certificate contains an **issuerUniqueID** or **subjectUniqueID** then the **version** field shall contain the integer **1** or **2**.

c) If the certificate contains **extensions** then the **version** field shall contain the integer **2**.

d) The **serialNumber** shall be unique with respect to the issuing Certification Authority.

e) The **validity** field shall specify a **notBefore** value that does not precede the current time and a **notAfter** value that does not precede the value specified in **notBefore**.

f) If the **issuer** field contains a null **Name** (e.g., a sequence of zero relative distinguished names), then the certificate shall contain a critical **issuerAltName** extension.

g) If the **subject** field contains a null **Name** (e.g., a sequence of zero relative distinguished names), then the certificate shall contain a critical **subjectAltName** extension.

h) The **signature** field and the **algorithm** in the **subjectPublicKeyInfo** field shall contain the OID for a FIPS-approved or recommended algorithm.

5.2.4.12 Certificate Revocation List Validation (FDP_CIMC_CRL.1)

FDP_CIMC_CRL.1.1 A TSF that issues CRLs shall verify that all mandatory fields in any CRL issued contain values in accordance with ITU-T Recommendation X.509. At a minimum, the following items shall be validated:

1. If the **version** field is present, then it shall contain a **2**[27].

2. If the CRL contains any critical extensions, then the version field shall be present and contain the integer **1**.

3. If the **issuer** field contains a null **Name** (e.g., a sequence of zero relative distinguished names), then the CRL shall contain a critical **issuerAltName** extension.

[27] While the source PP indicates a value of 1, there is now a subsequent version (i.e., 2) that is used by the TOE.

4. The signature and **signatureAlgorithm** fields shall contain the OID for a FIPS-approved digital signature algorithm.

5. The **thisUpdate** field shall indicate the issue date of the CRL.

6. The time specified in the **nextUpdate** field (if populated) shall not precede the time specified in the **thisUpdate** field.

5.2.4.13 Certificate Status Export (FDP_CIMC_CSE.1)

FDP_CIMC_CSE.1.1 Certificate status information shall be exported from the TOE in messages whose format complies with [**the X.509 standard for CRLs specified in RFC3280, except that the critical Issuing Distribution extension is not asserted in specific circumstances when the CRL does not cover certificates where the CA key signing the certificates is different from the CA key signing the CRL**].

5.2.4.14 OCSP Basic Response Validation (FDP_CIMC_OCSP.1)

FDP_CIMC_OCSP.1.1 If a TSF is configured to allow OCSP responses of the basic response type, the TSF shall verify that all mandatory fields in the OCSP basic response contain values in accordance with IETF RFC 2560. At a minimum, the following items shall be validated:

1. The **version** field shall contain a **0**.

2. If the **issuer** field contains a null **Name** (e.g., a sequence of zero relative distinguished names), then the response shall contain a critical **issuerAltName** extension.

3. The **signatureAlgorithm** field shall contain the OID for a FIPS-approved digital signature algorithm.

4. The **thisUpdate** field shall indicate the time at which the status being indicated is known to be correct.

5. The **producedAt** field shall indicate the time at which the OCSP responder signed the response.

6. The time specified in the **nextUpdate** field (if populated) shall not precede the time specified in the **thisUpdate** field.

5.2.4.15 Extended User Private and Secret Key Export (FDP_ETC_CIMC.5)

FDP_ETC_CIMC.5.1 Private and secret keys shall only be exported from the ~~TOE~~ **Active Directory Certificate Services role** in encrypted form or using split knowledge procedures. Electronically distributed secret and private keys shall only be exported from the TOE in encrypted form.

5.2.4.16 IPSec Subset Information Flow Control (FDP_IFC.1a)

FDP_IFC.1a.1 The TSF shall enforce the **[IPSec Filter Policy]** on **[**

a) subjects: one TSF sending IP traffic to another TSF or receiving IP traffic from another TSF,
b) information: IP traffic, and

c) **operation: pass information]**.

5.2.4.17 Windows Firewall Connection Subset Information Flow Control (FDP_IFC.1(b))

FDP_IFC.1b.1 The TSF shall enforce the **[Windows Firewall Connection Policy]** on **[**

a) subjects: one TSF receiving IP traffic from another TSF,
b) information: IP traffic, and
c) **operation: receive information]**.

5.2.4.18 IPSec Simple Security Attributes (FDP_IFF.1a)

FDP_IFF.1a.1 The TSF shall enforce the **[IPSec Filter Policy]** based on the following types of subject and information security attributes: **[**

a) subject security attributes: presumed address;
b) **information security attributes: presumed address of source subject, presumed address of destination subject, protocol, source port identification, and destination port identification]**.

FDP_IFF.1a.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold**: [all the information security attribute values are unambiguously permitted by the IPSec policy filter rules, where such rules may be composed from all possible combinations of the values of the information flow security attributes, created by the authorized administrator]**.

FDP_IFF.1a.3 The TSF shall enforce the **[no additional information control SFP rules]**.

FDP_IFF.1a.4 The TSF shall explicitly authorize an information flow based on the following rules: **[no explicit authorization rules]**.

FDP_IFF.1a.5 The TSF shall explicitly deny an information flow based on the following rules: **[no explicit deny rules]**.

5.2.4.19 Windows Firewall Connection Simple Security Attributes (FDP_IFF.1b)

FDP_IFF.1b.1 The TSF shall enforce the **[Windows Firewall Connection Policy]** based on the following types of subject and information security attributes: **[**

a) subject security attributes: Windows Firewall Connection Policy Port Mapping Rules;
b) Information security attributes: destination port identification].

FDP_IFF.1b.2 The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation if the following rules hold: **[the destination port is permitted by the Windows Firewall Connection Policy Port Mapping Rules]**.

FDP_IFF.1b.3 The TSF shall enforce the **[no additional information control SFP rules]**.

FDP_IFF.1b.4 The TSF shall explicitly authorize an information flow based on the following rules: **[the incoming packet is a response to previous outgoing packet or the Windows Firewall Connection Policy is not enabled]**.

FDP_IFF.1b.5 The TSF shall explicitly deny an information flow based on the following rules: **[no explicit deny rules]**.

5.2.4.20 Basic Internal Transfer Protection (FDP_ITT.1)

FDP_ITT.1.1 The TSF shall enforce the **[IPSec Filter Policy]** to prevent the ***[disclosure and modification]*** of user data when it is transmitted between physically-separated parts of the TOE.

5.2.4.21 Full Residual Information Protection (FDP_RIP.2)

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon ***[allocation to]*** all objects.

5.2.4.22 WEBUSER Basic Data Exchange Confidentiality (FDP_UCT.1)

FDP_UCT.1.1 The TSF shall enforce the **[WEBUSER SFP]** to ***[transmit and receive]*** user data in a manner protected from unauthorized disclosure.

5.2.4.23 WEBUSER SFP Data Exchange Integrity (FDP_UIT.1)

FDP_UIT.1.1 The TSF shall enforce the **[WEBUSER SFP]** to ***[transmit and receive]*** user data in a manner protected from ***[modification]*** errors.

FDP_UIT.1.2 The TSF shall be able to determine on receipt of user data, whether ***[modification]*** has occurred.

5.2.5 Identification and authentication (FIA)

5.2.5.1 Authentication Failure Handling (FIA_AFL_EXT.1)

FIA_AFL_EXT.1.1 The TSF shall detect when an authorized administrator configurable positive integer of consecutive unsuccessful authentication attempts occur related to any authorized user authentication process.

FIA_AFL_EXT.1.2 When the defined number of consecutive unsuccessful authentication attempts has been met or surpassed, the TSF shall:

a) For all administrator accounts, “disable” the account for an authorized administrator configurable time period such that there can be no more than ten attempts per minute.
b) For all other accounts, disable the user logon account until it is re-enabled by the authorized administrator.
c) For all disabled accounts, any response to an authentication attempt given to the user shall not be based on the result of that authentication attempt.

5.2.5.2 User Attribute Definition (FIA_ATD.1)

FIA_ATD.1.1 The TSF shall maintain the following list of security attributes belonging to individual users:

a) unique identifier,
b) group memberships,
c) authentication data,
d) security-relevant roles (see FMT_SMR.2),
e) **[Private/Public Keys]**, and
f) **[Privileges, and Logon Rights on specific physically separated parts of the TOE; Allowable time and day to logon; Policy requiring smart card to logon]**.

5.2.5.3 Verification of Secrets (FIA_SOS.1)

FIA_SOS.1.1 The TSF shall provide a mechanism to verify that secrets meet the following:

a) Passwords are at least 16 characters in length, consisting of any combination of upper and lower case letters, numbers, and symbols, and
b) Passwords are not reused within the last administrator-settable number of passwords used by that user.

5.2.5.4 Timing of Authentication (FIA_UAU.1)

FIA_UAU.1.1 The TSF shall allow read access to public objects on behalf of the user to be performed before the user is authenticated.

FIA_UAU.1.2 The TSF shall require each user to be successfully authenticated (i.e., an exact match between the internal representation of the user's entered data and the stored TSF authentication data) before allowing any other TSF-mediated actions on behalf of that user.

5.2.5.5 Re-authenticating (FIA_UAU.6)

FIA_UAU.6.1 The TSF shall re-authenticate the user when changing authentication data.

5.2.5.6 Protected Authentication Feedback (FIA_UAU.7)

FIA_UAU.7.1 The TSF shall provide only obscured feedback to the user while the authentication is in progress.

5.2.5.7 Timing of Identification (FIA_UID.1)

FIA_UID.1.1 The TSF shall allow read access to public objects on behalf of the user to be performed before the user is identified.

FIA_UID.1.2 The TSF shall require each user to be successfully identified before allowing any other TSF-mediated actions on behalf of that user.

5.2.5.8 User-Subject Binding (FIA_USB.1)

FIA_USB.1.1 The TSF shall associate the following user security attributes with subjects acting on behalf of that user: The security attributed identified in FIA_ATD.1 a, b, d, and **[FIA_ATD.1e when defined, and privileges and logon rights identified in FIA_ATD.1f]**.

FIA_USB.1.2 The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of users:

a) For administrative users, provide restrictive defaults for security attributes identified in FIA_ATD.1, ~~and~~

b) Restrict the ability to specify alternative initial user security attributes (that override the default attributes) to authorized administrators~~.~~**, and**

c) Mandatory Integrity Control integrity labels and policies are assigned as follows:

- **Subjects associated with non-administrative users receive a medium integrity level by default.**
- **Subjects associated with administrative users receive a high integrity level by default.**
- **Subjects started by another subject are assigned the lower of the integrity level assigned to the subject or the integrity level assigned to the executable file associated with the subject if they have the TOKEN_MANDATORY_POLICY_NEW_PROCESS_MIN mandatory policy configured; otherwise they are assigned the integrity level assigned to the executable file associated with the subject.**
- **All subjects are assigned the Mandatory Integrity Control policies: "TOKEN_MANDATORY_POLICY_NO_WRITE_UP" and "TOKEN_MANDATORY_POLICY_NEW_PROCESS_MIN" by default.**

FIA_USB.1.3 The TSF shall enforce the following rules governing changes to the user security attributes associated with subjects acting on the behalf of users:

a) User security attribute changes shall take effect at next user logon~~.~~**, and**

b) Subjects acting on behalf of users cannot add additional security attributes beyond those initially assigned, except when User Account Control is enabled in which case authorized administrators initially are assigned only access rights available to Standard Users and can subsequently escalate their access rights to their assigned (authorized administrator) level.

5.2.6 Security Management (FMT)

5.2.6.1 Management of Security Functions Behavior (for specification of auditable events) (FMT_MOF.1a)

FMT_MOF.1a.1 The TSF shall restrict the ability to disable and enable the audit functions and to specify which events are to be audited (see FAU_SEL.1.1) to the authorized administrators.

5.2.6.2 Management of Security Functions Behavior (for authentication data) (FMT_MOF.1b)

FMT_MOF.1b.1 The TSF shall restrict the ability to manage the values of security attributes associated with user authentication data to authorized administrators.

5.2.6.3 Management of Security Functions Behavior (FMT_MOF.1c)

FMT_MOF.1c.1 The TSF shall restrict the ability to modify the behavior of the functions listed in Table 3 to the authorized roles as specified in Table 3.

Table 3 Authorized Functions for Management of Security Functions Behavior (for Certificate Services)

Function	Authorized Role

Function	Authorized Role
Certificate Registration	The capability to approve fields or extensions to be included in a certificate shall be restricted to Officers.
Data Export and Output	The export of CIMC private keys shall require the authorization of at least two Administrators or one Administrator and one Officer, Auditor, or Operator.
Certificate Status Change Approval	Only Officers shall configure the process used to approve the revocation of a certificate or information about the revocation of a certificate. Only Officers shall configure the process used to approve the placing of a certificate on hold or information about the on hold status of a certificate.
CIMC Configuration	The capability to configure any TSF functionality shall be restricted to Administrators. (This requirement applies to all configuration parameters unless the ability to configure that aspect of the TSF functionality has been assigned to a different role elsewhere in this document.)
Certificate Profile Management	The capability to modify the certificate profile shall be restricted to Administrators.[28]
Revocation Profile Management	The capability to modify the revocation profile shall be restricted to Administrators.
Certificate Revocation List Profile Management	The capability to modify the certificate revocation list profile shall be restricted to Administrators.[29]
Online	The capability to modify the OCSP profile shall be restricted to Administrators.[30]

[28] i.e., FMT_MOF_CIMC.3 Extended certificate profile management

[29] i.e., FMT_MOF_CIMC.5 Extended certificate revocation List profile management

Function	Authorized Role
Certificate Status Protocol (OCSP) Profile Management	

Application Note: By default, these functions are performed by users assigned to the Administrators security group.

5.2.6.4 Extended Certificate Profile Management (FMT_MOF_CIMC.3)

FMT_MOF_CIMC.3.1 The TSF shall implement a certificate profile and shall ensure that issued certificates are consistent with that profile.

FMT_MOF_CIMC.3.2 The TSF shall require the Administrator to specify the set of acceptable values for the following fields and extensions:

- the key owner's identifier;
- the algorithm identifier for the subject’s public/private key pair;
- the identifier of the certificate issuer;
- the length of time for which the certificate is valid;

FMT_MOF_CIMC.3.3 If the certificates generated are X.509 public key certificates, the TSF shall require the Administrator to specify the set of acceptable values for the following fields and extensions:

- keyUsage;
- basicConstraints;
- certificatePolicies

FMT_MOF_CIMC.3.4 The Administrator shall specify the acceptable set of certificate extensions.

5.2.6.5 Extended Certificate Revocation List Profile Management (FMT_MOF_CIMC.5)

FMT_MOF_CIMC.5.1 If the TSF issues CRLs, the TSF must implement a certificate revocation list profile and ensure that issued CRLs are consistent with the certificate revocation list profile.

FMT_MOF_CIMC.5.2 If the TSF issues CRLs, the TSF shall require the Administrator to specify the set of acceptable values for the following fields and extensions:

- issuer;
- issuerAltName (NOTE: If a CIMC does not issue CRLs with this extension, then it is not required within the certificate revocation list profile.)

[30] i.e., FMT_MOF_CIMC.6 OCSP profile management

- nextUpdate (i.e., a promise of next CRL in specified time).

FMT_MOF_CIMC.5.3 If the TSF issues CRLs, the Administrator shall specify the acceptable set of CRL and CRL entry extensions.

5.2.6.6 OCSP Profile Management (FMT_MOF_CIMC.6)

FMT_MOF_CIMC.6.1 If the TSF issues OCSP responses, the TSF shall implement an OCSP profile and ensure that issued OCSP responses are consistent with the OCSP profile.

FMT_MOF_CIMC.6.2 If the TSF issues OCSP responses, the TSF shall require the Administrator to specify the set of acceptable values for the **responseType** field (unless the CIMC can only issue responses of the basic response type).

FMT_MOF_CIMC.6.3 If the TSF is configured to allow OCSP responses of the basic response type, the TSF shall require the Administrator to specify the set of acceptable values for the **ResponderID** field within the basic response type.

5.2.6.7 Management of Security Attributes (for discretionary access control) (FMT_MSA.1a)

FMT_MSA.1a.1 The TSF shall enforce the Discretionary Access Control policy to restrict the ability to change the value of object security attributes to authorized administrators, owners of the object **[and no other rules]**.

5.2.6.8 Management of Security Attributes (for object ownership) (FMT_MSA.1b)

FMT_MSA.1b.1 The TSF shall enforce the Discretionary Access Control policy to restrict the ability to change object ownership to authorized administrators.

5.2.6.9 Management of IPSec Object Security Attributes (FMT_MSA.1c)

FMT_MSA.1c.1 The TSF shall enforce the **[IPSec Filter Policy]** to restrict the ability to ***[modify]*** the security attributes **[IPSec Filter Policy security attributes]** to **[the authorized administrator]**.

5.2.6.10 Management of Windows Firewall Connection Object Security Attributes (FMT_MSA.1d)

FMT_MSA.1d.1 The TSF shall enforce the **[Windows Firewall Connection Policy]** to restrict the ability to ***[modify]*** the security attributes **[Windows Firewall Connection Policy security attributes]** to **[the authorized administrator]**.

5.2.6.11 Management of WEBUSER Object Security Attributes (FMT_MSA.1e)

FMT_MSA.1e.1 The TSF shall enforce the **[WEBUSER SFP]** to restrict the ability to ***[modify]*** the security attributes **[WEBUSER SFP security attributes]** to **[the authorized administrator]**.

5.2.6.12 Management of CONTENT-PROVIDER Object Security Attributes (FMT_MSA.1f)

FMT_MSA.1f.1 The TSF shall enforce the **[CONTENT-PROVIDER SFP]** to restrict the ability to ***[modify]*** the security attributes **[CONTENT-PROVIDER SFP security attributes]** to **[the authorized administrator]**.

5.2.6.13 Management of Mandatory Integrity Control Security Attributes (FMT_MSA.1g)

FMT_MSA.1g.1 The TSF shall enforce the **[Mandatory Integrity Control Policy]** to restrict the ability to ***[modify]*** the security attributes **[integrity labels]** to **[the authorized administrator]**.

5.2.6.14 Secure Security Attributes (FMT_MSA.2)

FMT_MSA.2.1 The TSF shall ensure that only valid values are accepted for all security attributes.

5.2.6.15 Static Attribute Initialization (FMT_MSA.3a)

FMT_MSA.3a.1 The TSF shall enforce the Discretionary Access Control policy to provide restrictive default values for security attributes that are used to enforce the SFP.

FMT_MSA.3a.2 The TSF shall allow the authorized administrator to specify alternative initial values to override the default values when an object or information is created.

5.2.6.16 IPSec Static Attribute Initialization (FMT_MSA.3b)

FMT_MSA.3b.1 The TSF shall enforce the **[IPSec Filter Policy]** to provide ***[permissive]*** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3b.2 The TSF shall allow the **[creator or authorized administrator]** to specify alternative initial values to override the default values when an object or information is created.

5.2.6.17 Windows Firewall Connection Static Attribute Initialization (FMT_MSA.3c)

FMT_MSA.3c.1 The TSF shall enforce the **[Windows Firewall Connection Policy]** to provide ***[[permissive for Server 2008 R2 and restrictive for Windows 7]]*** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3c.2 The TSF shall allow the **[authorized administrator]** to specify alternative initial values to override the default values when an object or information is created.

5.2.6.18 WEBUSER Static Attribute Initialization (FMT_MSA.3d)

FMT_MSA.3d.1 The TSF shall enforce the **[WEBUSER SFP]** to provide ***[restrictive]*** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3d.2 The TSF shall allow the **[authorized administrator]** to specify alternative initial values to override the default values when an object or information is created.

5.2.6.19 CONTENT-PROVIDER Static Attribute Initialization (FMT_MSA.3e)

FMT_MSA.3e.1 The TSF shall enforce the **[CONTENT-PROVIDER SFP]** to provide ***[restrictive]*** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3e.2 The TSF shall allow the **[authorized administrator]** to specify alternative initial values to override the default values when an object or information is created.

5.2.6.20 Mandatory Integrity Attribute Initialization (FMT_MSA.3f)

FMT_MSA.3f.1 The TSF shall enforce the **[Mandatory Integrity Control Policy]** to provide ***[restrictive]*** default values for security attributes that are used to enforce the SFP.

FMT_MSA.3f.2 The TSF shall allow the **[authorized administrator]** to specify alternative initial values to override the default values when an object or information is created.

5.2.6.21 Management of TSF Data (for general TSF data) (FMT_MTD.1a)

FMT_MTD.1a.1 The TSF shall restrict the ability to manage the TSF data except for audit records, user security attributes, authentication data, and critical cryptographic security parameters to authorized administrators.

5.2.6.22 Management of TSF Data (for audit data) (FMT_MTD.1b)

FMT_MTD.1b.1 The TSF shall restrict the ability to query, delete, and clear the audit records to authorized administrators.

5.2.6.23 Management of TSF Data (for initialization of user security attributes) (FMT_MTD.1c)

FMT_MTD.1c.1 The TSF shall restrict the ability to initialize user security attributes to authorized administrators.

5.2.6.24 Management of TSF Data (for modification of user security attributes, other than authentication data) (FMT_MTD.1d)

FMT_MTD.1d.1 The TSF shall restrict the ability to modify user security attributes, other than authentication data, to authorized administrators.

5.2.6.25 Management of TSF Data (for modification of authentication data) (FMT_MTD.1e)

FMT_MTD.1e.1 The TSF shall restrict the ability to modify authentication data to authorized administrators and users modifying their own authentication data.

5.2.6.26 Management of TSF Data (for reading of authentication data) (FMT_MTD.1f)

FMT_MTD.1f.1 The TSF shall prevent reading of authentication data.

5.2.6.27 Management of TSF Data (for critical cryptographic security parameters) (FMT_MTD.1g)

FMT_MTD.1g.1 The TSF shall restrict the ability to manage the critical cryptographic security parameters and data related to cryptographic configuration to authorized administrators.

5.2.6.28 TSF Private Key Confidentiality Protection (FMT_MTD_CIMC.4)

FMT_MTD_CIMC.4.1 **TOE** ~~CIMC~~ private keys shall be stored in a FIPS 140-2 validated cryptographic module or stored in encrypted form. If **TOE** ~~CIMC~~ private keys are stored in encrypted form, the encryption shall be performed by the FIPS 140-2 validated cryptographic module.

5.2.6.29 TSF Secret Key Confidentiality Protection (FMT_MTD_CIMC.5)

FMT_MTD_CIMC.5.1 TSF secret keys stored within the TOE, but not within a FIPS 140-2 validated cryptographic module, shall be stored in encrypted form. The encryption shall be performed by the FIPS 140-2 validated cryptographic module.

5.2.6.30 Extended TSF Private and Secret Key Export (FMT_MTD_CIMC.7)

FMT_MTD_CIMC.7.1 Private and secret keys shall only be exported from the TOE in encrypted form or using split knowledge procedures. Electronically distributed secret and private keys shall only be exported from the TOE in encrypted form.

5.2.6.31 Revocation (to authorized administrators) (FMT_REV.1a)

FMT_REV.1a.1 The TSF shall restrict the ability to revoke security attributes associated with the users under the control of the TSF to authorized administrators.

FMT_REV.1a.2 The TSF shall enforce the revocation of security-relevant authorizations at the next logon.

5.2.6.32 Revocation (to owners and authorized administrators) (FMT_REV.1b)

FMT_REV.1b.1 The TSF shall restrict the ability to revoke security attributes of named objects to owners of the named object and authorized administrators.

FMT_REV.1b.2 The TSF shall enforce the revocation of access rights associated with named objects when an access check is made.

5.2.6.33 Time-Limited Authorization (FMT_SAE.1)

FMT_SAE.1.1 The TSF shall restrict the capability to specify an expiration time for authorized user authentication data to the authorized administrator.

FMT_SAE.1.2 The TSF shall be able to force the associated authorized user to change their authentication information prior to being able to successfully log on after the expiration time has passed.

5.2.6.34 Specification of Management Functions (FMT_SMF.1)

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: all security management functions identified in other sections of this ~~PP~~**ST**.

5.2.6.35 Security Roles (FMT_SMR.1)

FMT_SMR.1.1 The TSF shall maintain the roles:

a) authorized administrator,
b) [no other roles].

FMT_SMR.1.2 The TSF shall be able to associate authorized users with roles.

5.2.7 Protection of the TSF (FPT)

5.2.7.1 Basic internal TSF Data Transfer Protection (FPT_ITT.1)

FPT_ITT.1.1 The TSF shall protect TSF data from disclosure when it is transmitted between separate parts of the TOE through the use of the TSF-provided cryptographic services: **[encryption and decryption]**.

5.2.7.2 TSF Data Integrity Monitoring (FPT_ITT.3)

FPT_ITT.3.1 The TSF shall be able to detect modification and insertion of TSF data transmitted between separate parts of the TOE through the use of the TSF-provided cryptographic services: **[cryptographic signature and cryptographic hashing]**.

FPT_ITT.3.2 Upon detection of a data integrity error, the TSF shall take the following actions:

a) audit event, and
b) **[reject data]**.

5.2.7.3 Manual Recovery (FPT_RCV.1)

FPT_RCV.1.1 After a failure or service discontinuity that may lead to a violation of the TSP, the TSF shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.

5.2.7.4 TSF Hardware Protection (FPT_WPF_EX.1)

FPT_WPF_EX.1.1 The TSF in 64-bit architectures shall allow a subject to choose an option whereby the TSF shall prevent the subject from executing data on a memory page that is not marked for execution.

FPT_WPF_EX.1.2 The TSF shall prevent a subject from executing data on a memory page that is not marked for execution after the subject has selected such an option.

5.2.7.5 TSF Disk Volume Protection (FPT_WPF_EX.2)

FPT_WPF_EX.2.1 The TSF shall be able to protect the persistent representation of itself, TSF data, and user data from modification and disclosure while the TSF is stopped.

FPT_WPF_EX.2.2 The TSF shall be able to require entry of appropriate credentials in order to access the TSF, TSF data, and user data in order to start the TSF or recover protected data.

5.2.7.6 Removable USB Storage Device Protection (FPT_WPF_EX.3)

FPT_WPF_EX.3.1 The TSF shall be able to protect the content of removable USB storage devices from modification and disclosure.

FPT_WPF_EX.3.2 The TSF shall allow access to protected content on removable USB storage devices only after valid credentials are provided.

FPT_WPF_EX.3.3 The TSF shall be able to require that removable USB storage device content can only be written in a form that protects the data from modification and disclosure; otherwise the content of such devices can only be read and not written.

5.2.7.7 Network Access Protection (FPT_WPF_EX.4)

FPT_WPF_EX.4.1 The TSF shall provide only an authorized administrator with the capability to define a minimum NAP Health Policy.

FPT_WPF_EX.4.2 The TSF shall be able to measure the NAP Health of a client TOE component in terms of the following attributes: TSF data and installed applications.

FPT_WPF_EX.4.3 The TSF shall be able to compare the NAP Health of a client against the configured NAP Health Policy to determine compliance or non-compliance.

FPT_WPF_EX.4.4 The TSF shall provide access to authorized administrator-configured network access credentials only when a client's NAP Health is in compliance with the NAP Health Policy.

5.2.7.8 TPM Full Volume Encryption Support (FPT_WPF_EX.5)

FPT_WPF_EX.5.1 The TSF shall be able to utilize the services of an attached TPM chip to store and retrieve the keys used for the purpose of Full Volume Encryption.

FPT_WPF_EX.5.2 The TSF shall be able to utilize the services of an attached TPM chip to withhold the Full Volume Encryption keys until the integrity of a subset of the operating system components is confirmed.

5.2.7.9 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

5.2.7.10 Internal TSF Data Consistency (FPT_TRC_EXT.1)

FPT_TRC_EXT.1.1 The TSF shall ensure that TSF data is consistent between parts of the TOE by providing a mechanism to bring inconsistent TSF data into a consistent state without undue delay.

5.2.7.11 TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1 The TSF shall run a suite of self-tests in accordance with FIPS PUB 140-2 during initial start-up (on power on) to demonstrate the correct operation of the cryptographic modules.

FPT_TST_EXT.1.2 The TSF shall provide the capability to verify the integrity of stored TSF executable code through the use of the TSF-provided cryptographic services.

FPT_TST_EXT.1.3 The TSF shall verify the integrity of the following TSF data: authentication data, **[and all other TSF data]** at start up.

5.2.8 Resource utilization (FRU)

5.2.8.1 Maximum Quotas (FRU_RSA.1)

FRU_RSA.1.1 The TSF shall enforce maximum quotas of the following resources: portion of shared persistent storage that individual authorized users can use simultaneously.

5.2.9 TOE access (FTA)

5.2.9.1 Basic Limitation on Multiple Concurrent Sessions (FTA_MCS.1)

FTA_MCS.1.1 The TSF shall enforce a maximum number of concurrent interactive sessions per user.

FTA_MCS.1.2 The TSF shall allow an authorized administrator to set the maximum number of concurrent interactive sessions per user.

5.2.9.2 TSF-initiated Session Locking (FTA_SSL.1)

FTA_SSL.1.1 The TSF shall lock an interactive session after an authorized administrator specified time interval of user inactivity by:

a) clearing or overwriting display devices, making the current contents unreadable.
b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.1.2 The TSF shall require the user to re-authenticate to unlock the session.

5.2.9.3 User-initiated Locking (FTA_SSL.2)

FTA_SSL.2.1 The TSF shall allow user-initiated locking of the user's own interactive session by:

a) clearing or overwriting display devices, making the current contents unreadable.
b) disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.2.2 The TSF shall require the user to re-authenticate to unlock the session.

5.2.9.4 WEBUSER TSF-Initiated Termination (FTA_SSL.3)

FTA_SSL.3.1 The TSF shall terminate a~~n~~ remote interactive **http:// or https://** session after a **[an administrator configurable time interval of session inactivity]**.

5.2.9.5 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1 Before establishing a user session, the TSF shall display an authorized-administrator specified advisory notice and consent warning message regarding unauthorized use of the TOE.

5.2.9.6 TOE Access History (FTA_TAH.1)

FTA_TAH.1.1 Upon successful interactive session establishment, the TSF shall display to the authorized user the date and time of that authorized user's last successful interactive session establishment.

FTA_TAH.1.2 Upon successful interactive session establishment, the TSF shall display to the authorized user the date and time of the last unsuccessful attempt and the number of unsuccessful attempts at interactive session establishment for that user identifier since the last successful interactive session establishment.

FTA_TAH.1.3 The TSF shall not erase the access history information from the authorized user interface without giving the authorized user the opportunity to review the information.

5.2.9.7 TOE Session Establishment (FTA_TSE.1)

FTA_TSE.1.1 The TSF shall be able to deny session establishment based on **[authentication data expiration, location, time, and day]**.

5.2.10 Trusted Path/Channels

5.2.10.1 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1 The TSF shall provide a communication path between itself and ***[local]*** users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from ***[modification or disclosure]***.

FTP_TRP.1.2 The TSF shall permit ***[local users]*** to initiate the communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for ***[initial user authentication with password, initial user authentication with smartcard, change password, and session unlocking]***.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the EAL 4 augmented with ALC_FLR.3 components as specified in Part 3 of the Common Criteria. No operations are applied to the assurance components.

Table 5-4 TOE Security Assurance Requirements

Requirement Class	Requirement Component
ADV: Development	ADV_ARC.1: Security architecture description
	ADV_FSP.4: Complete functional specification
	ADV_IMP.1: Implementation representation of the TSF
	ADV_TDS.3: Basic modular design
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.4: Production support, acceptance procedures and automation
	ALC_CMS.4: Problem tracking CM coverage
	ALC_DEL.1: Delivery procedures
	ALC_DVS.1: Identification of security measures
	ALC_FLR.3: Systematic flaw remediation
	ALC_LCD.1: Developer defined life-cycle model
	ALC_TAT.1: Well-defined development tools
ATE: Tests	ATE_COV.2: Analysis of coverage
	ATE_DPT.1: Testing: basic design
	ATE_FUN.1: Functional testing
	ATE_IND.2: Independent testing - sample
AVA: Vulnerability assessment	AVA_VAN.3: Focused vulnerability analysis

5.3.1 Development (ADV)

5.3.1.1 Security Architecture Description (ADV_ARC.1)

ADV_ARC.1.1d The developer shall design and implement the TOE so that the security features of the TSF cannot be bypassed.

ADV_ARC.1.2d The developer shall design and implement the TSF so that it is able to protect itself from tampering by untrusted active entities.

ADV_ARC.1.3d The developer shall provide a security architecture description of the TSF.

ADV_ARC.1.1c The security architecture description shall be at a level of detail commensurate with the description of the SFR-enforcing abstractions described in the TOE design document.

ADV_ARC.1.2c The security architecture description shall describe the security domains maintained by the TSF consistently with the SFRs.

ADV_ARC.1.3c The security architecture description shall describe how the TSF initialisation process is secure.

ADV_ARC.1.4c The security architecture description shall demonstrate that the TSF protects itself from tampering.

ADV_ARC.1.5c The security architecture description shall demonstrate that the TSF prevents bypass of the SFR-enforcing functionality.

ADV_ARC.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.1.2 Complete Functional Specification (ADV_FSP.4)

ADV_FSP.4.1d The developer shall provide a functional specification.

ADV_FSP.4.2d The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.4.1c The functional specification shall completely represent the TSF.

ADV_FSP.4.2c The functional specification shall describe the purpose and method of use for all TSFI.

ADV_FSP.4.3c **T**he functional specification shall identify and describe all parameters associated with each TSFI.

ADV_FSP.4.4c The functional specification shall describe all actions associated with each TSFI.

ADV_FSP.4.5c The functional specification shall describe all direct error messages that may result from an invocation of each TSFI.

ADV_FSP.4.6c The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.4.2e The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.3.1.3 Implementation Representation of the TSF (ADV_IMP.1)

ADV_IMP.1.1d The developer shall make available the implementation representation for the entire TSF.

ADV_IMP.1.2d The developer shall provide a mapping between the TOE design description and the sample of the implementation representation.

ADV_IMP.1.1c The implementation representation shall define the TSF to a level of detail such that the TSF can be generated without further design decisions.

ADV_IMP.1.2c The implementation representation shall be in the form used by the development personnel.

ADV_IMP.1.3c The mapping between the TOE design description and the sample of the implementation representation shall demonstrate their correspondence.

ADV_IMP.1.1e The evaluator shall confirm that, for the selected sample of the implementation representation, the information provided meets all requirements for content and presentation of evidence.

5.3.1.4 Basic Modular Design (ADV_TDS.3)

ADV_TDS.3.1d The developer shall provide the design of the TOE.

ADV_TDS.3.2d The developer shall provide a mapping from the TSFI of the functional specification to the lowest level of decomposition available in the TOE design.

ADV_TDS.3.1c The design shall describe the structure of the TOE in terms of subsystems.

ADV_TDS.3.2c The design shall describe the TSF in terms of modules.

ADV_TDS.3.3c The design shall identify all subsystems of the TSF.

ADV_TDS.3.4c The design shall provide a description of each subsystem of the TSF.

ADV_TDS.3.5c The design shall provide a description of the interactions among all subsystems of the TSF.

ADV_TDS.3.6c The design shall provide a mapping from the subsystems of the TSF to the modules of the TSF.

ADV_TDS.3.7c The design shall describe each SFR-enforcing module in terms of its purpose and relationship with other modules.

ADV_TDS.3.8c The design shall describe each SFR-enforcing module in terms of its SFR-related interfaces, return values from those interfaces, interaction with other modules and called SFR-related interfaces to other SFR-enforcing modules.

ADV_TDS.3.9c The design shall describe each SFR-supporting or SFR-non-interfering module in terms of its purpose and interaction with other modules.

ADV_TDS.3.10c The mapping shall demonstrate that all TSFIs trace to the behaviour described in the TOE design that they invoke.

ADV_TDS.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_TDS.3.2e The evaluator shall determine that the design is an accurate and complete instantiation of all security functional requirements.

5.3.2 Guidance documents (AGD)

5.3.2.1 Operational User Guidance (AGD_OPE.1)

AGD_OPE.1.1d The developer shall provide operational user guidance.

AGD_OPE.1.1c The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6c The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1d The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1c The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2e The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.3.3 Life-cycle support (ALC)

5.3.3.1 Production Support, Acceptance Procedures and Automation (ALC_CMC.4)

ALC_CMC.4.1d The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.4.2d The developer shall provide the CM documentation.

ALC_CMC.4.3d The developer shall use a CM system.

ALC_CMC.4.1c The TOE shall be labelled with its unique reference.

ALC_CMC.4.2c The CM documentation shall describe the method used to uniquely identify the configuration items.

ALC_CMC.4.3c The CM system shall uniquely identify all configuration items.

ALC_CMC.4.4c The CM system shall provide automated measures such that only authorised changes are made to the configuration items.

ALC_CMC.4.5c The CM system shall support the production of the TOE by automated means.

ALC_CMC.4.6c The CM documentation shall include a CM plan.

ALC_CMC.4.7c The CM plan shall describe how the CM system is used for the development of the TOE.

ALC_CMC.4.8c The CM plan shall describe the procedures used to accept modified or newly created configuration items as part of the TOE.

ALC_CMC.4.9c The evidence shall demonstrate that all configuration items are being maintained under the CM system.

ALC_CMC.4.10c The evidence shall demonstrate that the CM system is being operated in accordance with the CM plan.

ALC_CMC.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.2 Problem Tracking CM Coverage (ALC_CMS.4)

ALC_CMS.4.1d The developer shall provide a configuration list for the TOE.

ALC_CMS.4.1c The configuration list shall include the following: the TOE itself; the evaluation evidence required by the SARs; the parts that comprise the TOE; the implementation representation; and security flaw reports and resolution status.

ALC_CMS.4.2c The configuration list shall uniquely identify the configuration items.

ALC_CMS.4.3c For each TSF relevant configuration item, the configuration list shall indicate the developer of the item.

ALC_CMS.4.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.3 Delivery Procedures (ALC_DEL.1)

ALC_DEL.1.1d The developer shall document and provide procedures for delivery of the TOE or parts of it to the consumer.

ALC_DEL.1.2d The developer shall use the delivery procedures.

ALC_DEL.1.1c The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to the consumer.

ALC_DEL.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.4 Identification of Security Measures (ALC_DVS.1)

ALC_DVS.1.1d The developer shall produce and provide development security documentation.

ALC_DVS.1.1c The development security documentation shall describe all the physical, procedural, personnel, and other security measures that are necessary to protect the confidentiality and integrity of the TOE design and implementation in its development environment.

ALC_DVS.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ALC_DVS.1.2e The evaluator shall confirm that the security measures are being applied.

5.3.3.5 Systematic Flaw Remediation (ALC_FLR.3)

ALC_FLR.3.1d The developer shall document and provide flaw remediation procedures addressed to TOE developers.

ALC_FLR.3.2d The developer shall establish a procedure for accepting and acting upon all reports of security flaws and requests for corrections to those flaws.

ALC_FLR.3.3d The developer shall provide flaw remediation guidance addressed to TOE users.

ALC_FLR.3.1c The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the TOE.

ALC_FLR.3.2c The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.

ALC_FLR.3.3c The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.

ALC_FLR.3.4c The flaw remediation procedures documentation shall describe the methods used to provide flaw information, corrections and guidance on corrective actions to TOE users.

ALC_FLR.3.5c The flaw remediation procedures shall describe a means by which the developer receives from TOE users reports and enquiries of suspected security flaws in the TOE.

ALC_FLR.3.6c The flaw remediation procedures shall include a procedure requiring timely response and the automatic distribution of security flaw reports and the associated corrections to registered users who might be affected by the security flaw.

ALC_FLR.3.7c The procedures for processing reported security flaws shall ensure that any reported flaws are remediated and the remediation procedures issued to TOE users.

ALC_FLR.3.8c The procedures for processing reported security flaws shall provide safeguards that any corrections to these security flaws do not introduce any new flaws.

ALC_FLR.3.9c The flaw remediation guidance shall describe a means by which TOE users report to the developer any suspected security flaws in the TOE.

ALC_FLR.3.10c The flaw remediation guidance shall describe a means by which TOE users may register with the developer, to be eligible to receive security flaw reports and corrections.

ALC_FLR.3.11c The flaw remediation guidance shall identify the specific points of contact for all reports and enquiries about security issues involving the TOE.

ALC_FLR.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence

5.3.3.6 Developer Defined Life-cycle Model (ALC_LCD.1)

ALC_LCD.1.1d The developer shall establish a life-cycle model to be used in the development and maintenance of the TOE.

ALC_LCD.1.2d The developer shall provide life-cycle definition documentation.

ALC_LCD.1.1c The life-cycle definition documentation shall describe the model used to develop and maintain the TOE.

ALC_LCD.1.2c The life-cycle model shall provide for the necessary control over the development and maintenance of the TOE.

ALC_LCD.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.3.7 Well-defined Development Tools (ALC_TAT.1)

ALC_TAT.1.1d The developer shall provide the documentation identifying each development tool being used for the TOE.

ALC_TAT.1.2d The developer shall document and provide the selected implementation-dependent options of each development tool.

ALC_TAT.1.1c Each development tool used for implementation shall be well-defined.

ALC_TAT.1.2c The documentation of each development tool shall unambiguously define the meaning of all statements as well as all conventions and directives used in the implementation.

ALC_TAT.1.3c The documentation of each development tool shall unambiguously define the meaning of all implementation-dependent options.

ALC_TAT.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Tests (ATE)

5.3.4.1 Analysis of Coverage (ATE_COV.2)

ATE_COV.2.1d The developer shall provide an analysis of the test coverage.

ATE_COV.2.1c The analysis of the test coverage shall demonstrate the correspondence between the tests in the test documentation and the TSFIs in the functional specification.

ATE_COV.2.2c The analysis of the test coverage shall demonstrate that all TSFIs in the functional specification have been tested.

ATE_COV.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.2 Testing: Basic Design (ATE_DPT.1)

ATE_DPT.1.1d The developer shall provide the analysis of the depth of testing.

ATE_DPT.1.1c The analysis of the depth of testing shall demonstrate the correspondence between the tests in the test documentation and the TSF subsystems in the TOE design.

ATE_DPT.1.2c The analysis of the depth of testing shall demonstrate that all TSF subsystems in the TOE design have been tested.

ATE_DPT.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.3 Functional Testing (ATE_FUN.1)

ATE_FUN.1.1d The developer shall test the TSF and document the results.

ATE_FUN.1.2d The developer shall provide test documentation.

ATE_FUN.1.1c The test documentation shall consist of test plans, expected test results and actual test results.

ATE_FUN.1.2c The test plans shall identify the tests to be performed and describe the scenarios for performing each test. These scenarios shall include any ordering dependencies on the results of other tests.

ATE_FUN.1.3c The expected test results shall show the anticipated outputs from a successful execution of the tests.

ATE_FUN.1.4c The actual test results shall be consistent with the expected test results.

ATE_FUN.1.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4.4 Independent Testing - Sample (ATE_IND.2)

ATE_IND.2.1d The developer shall provide the TOE for testing.

ATE_IND.2.1c The TOE shall be suitable for testing.

ATE_IND.2.2c The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.

ATE_IND.2.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.2.2e The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

ATE_IND.2.3e The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.3.5 Vulnerability Assessment (AVA)

5.3.5.1 Focused Vulnerability Analysis (AVA_VAN.3)

AVA_VAN.3.1d The developer shall provide the TOE for testing.

AVA_VAN.3.1c The TOE shall be suitable for testing.

AVA_VAN.3.1e The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AVA_VAN.3.2e The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.3.3e The evaluator shall perform an independent, focused vulnerability analysis of the TOE using the guidance documentation, functional specification, TOE design, security architecture description and implementation representation to identify potential vulnerabilities in the TOE.

AVA_VAN.3.4e The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Enhanced-Basic attack potential.

6 TOE Summary Specification (TSS)

This chapter describes the Windows 7 and Windows Server 2008 R2 security functions. The Windows 7 and Windows Server 2008 R2 Security Functions (SFs) satisfy the security functional requirements of the GPOSPP. The TOE also includes additional SFs. The SFs relevant to Windows 7 and Windows Server 2008 R2 are also described in the following sections, as well as a mapping to the security functional requirements satisfied by the TOE.

6.1 TOE Security Functions

This section presents the TOE Security Functions (TSFs) and a mapping of security functions to Security Functional Requirements (SFRs). The TOE performs the following security functions:

- Audit
- User Data Protection
- Cryptographic Protection
- Identification and Authentication
- Security Management
- TSF Protection
- Resource Utilization
- TOE Access

6.1.1 Audit Function

The TOE Audit security function performs:

- Audit Collection
- Audit Log Review
- Selective Audit
- Audit Log Overflow Protection
- Audit Log Restricted Access Protection

6.1.1.1 Audit Collection

The Event logger service creates the security event log, which contains the security relevant audit records collected on a system. There is one security log (audit log) per machine. The Local Security Authority (LSA) server collects audit events from all other parts of the TSF and forwards them to the Event Logger for storage in the security log. For each audit event, the Event Logger stores the following data in each audit record:

Date:	The date the event occurred.
Time:	The time the event occurred.

User: The security identifier (SID) of that represents the user on whose behalf the event occurred that represents the user. SIDs are described in more detail in Section 6 under Identification and Authentication.

Event ID: A unique number identifying the particular event class.

Source: The system restricts what processes are capable of writing events to the security event log.

Outcome: Indicates whether the security audit event recorded is the result of a successful or failed attempt to perform the action.

Category: The type of the event defined by the event source. For the security log, the LSA service defines the following categories for security audit events: System, Logon, Object Access, Privilege Use, Detailed Process Tracking, Policy Change, Account Management, Directory Service Access, and Account Logon.

Each audit event may also contain category-specific data that is contained in the body of the event as described below:

- For the System Category, the audit records additionally include information relating to the system such as the time the audit trail was cleared, start or shutdown of the audit function, and startup and shutdown of Windows. Furthermore, the specific cryptographic operation is identified when such operations are audited.
- For the Object Access and the Directory Service Access Category, the audit records additionally include the object name and the desired access requested.
- For the Privilege Use Category, the audit records additionally identify the privilege.
- For the Detailed Process Tracking Category, the audit records additionally include the process identifier.
- For the Policy Change and Account Management Category, the audit records additionally include new values of the policy or account attributes.
- For the Logon and Account Logon Category, the audit records additionally include the reason for failure of attempted logons.
- For the Logon Category, the audit records additionally include the logon type that indicates the source of the logon attempt as one of the following types in the audit record:
    - Interactive (local logon)
    - Network (logon from the network)
    - Service (logon as a service)
    - Batch (logon as a batch job)
    - Unlock (for Unlock screen saver)
    - Network_ClearText (for anonymous authentication to IIS)

> **Note:** In the evaluated configuration IIS will only accept requests from authenticated clients, however, if configured for anonymous authentication IIS will not force the user to re-authenticate themselves and a specified account (identified by the authorized administrator) will be associated with the user.

There are two places within the TSF where security audit events are collected. The Security Reference Monitor (SRM) is responsible for the generation of all audit records for the object access, privilege use, and detailed process tracking event categories. With one exception, audit events for the remainder of the event categories are generated by various services that co-exist in the security process within the LSA server or call the Authz Report Audit APIs provided by the LSA Policy subcomponent. The exception is that the Event Logger itself records an event record when the security log is cleared and when the security log exceeds the warning level configured by the authorized administrator.

The LSA server maintains an audit policy in its database that determines which categories of events are actually collected. Defining and modifying the audit policy is restricted to the authorized administrator. The authorized administrator can select events to be audited by selecting the category or categories to be audited. An authorized administrator can individually select each category. Those services in the security process can determine the current audit policy via direct local function calls. The only other TSF component that uses the audit policy is the SRM in order to control object access, privilege use, and detailed tracking audit. LSA and the SRM share a private local connection port, which is used to pass the audit policy to the SRM. When an authorized administrator changes the audit policy, the LSA updates its database and notifies the SRM. The SRM receives a control flag indicating if auditing is enabled and a data structure indicating that the events in particular categories will be audited.

In addition to the system-wide audit policy configuration, it is possible to define a per-user audit policy. This allows individual audit categories (of success or failure) to be enabled or disabled on a per user basis. The per-user audit policy refines the system-wide audit policy by defining a more precise definition of the audit policy for which events will be masked and/or audited for a specific user.

Within each category, auditing can be performed based on success, failure, or both. For object access events, auditing can be further controlled based on user/group identify and access rights using System Access Control Lists (SACLs). SACLs are associated with objects and indicate whether or not auditing for a specific object, or object attribute, is enabled.

The TSF is capable of generating the audit events associated with each audit category, as described in the Description column of **Table 6-1** (**Audit Event Categories**). The auditable events associated with each category capture the events listed in **Table 5-2**. For each category, the associated audit events (listed in **Table 5-2**) for each of the requirements in the FAU_GEN Required Events column of **Table 6-1** are captured.

Table 6-1 Audit Event Categories

Category	Description	FAU_GEN Required Events
System	Audit attempts that affect security of the entire system such as clearing the audit trail.	FAU_STG.3, FCS_BCM_EXT.1, FCS_CKM.1, FCS_CKM.1, FCS_CKM.4, FCS_COP.1*, FCS_RBG_EXT.1, FMT_MTD.1a, FMT_MTD.1g, FMT_SMF.1, FPT_STM.1, FPT_RCV.1, FPT_WPF_EX.4, FPT_TST.1, FDP_CIMC_CER.1, FDP_CIMC_CRL.1, FDP_CIMC_OCSP.1
Object Access	Audit attempts to access user objects, such as files.	FDP_ACF.1*, FDP_IFF.1*, FDP_ITT.1, FDP_UCT.1, FDP_UIT.1, FMT_MSA.1a, FMT_MSA.1b, FMT_MSA.3a, FMT_REV.1b, FPT_WPF_EX.3, FRU_RSA.1
Privilege Use	Audits attempts to use security relevant privileges. Security relevant privileges are those privileges that are related to the TSFs and can be assigned in the evaluated configuration.	FAU_SAR.1, FAU_SAR.2, FDP_ACF.1a, FMT_SMF.1, FMT_SMR.1
Detailed Process Tracking	Audit subject-tracking events, including program activation, handle duplication, indirect access to an object, and process exit.	FIA_USB.1
Policy Change	Audit attempts to change security policy settings such as the audit policy and privilege assignment.	FAU_SEL.1, FMT_MOF.1*, FMT_MTD.1a, FMT_MTD.1b, FMT_SMF.1, FMT_REV.1a, FMT_SMR.1, FPT_ITT.3, FMT_MOF_CIMC.3, FMT_MOF_CIMC.6
Account Management	Audit attempts to create, delete, or change user or group accounts and changes to their attributes.	FIA_AFL_EXT.1, FMT_MSA.2, FMT_MTD.1c, FMT_MTD.1d, FMT_MTD.1e, FMT_REV1.a, FMT_SAE.1, FMT_SMF.1, FMT_SMR.1
Directory Service Access	Audit access to directory service objects and associated properties.	FDP_ACF.1a
Logon	Audit attempts to logon or logoff the system, attempts to make a network connection.	FIA_AFL.1, FIA_SOS.1, FIA_UAU.1, FIA_UAU.6, FIA_UID.1, FIA_USB.1, FTA_SSL.1, FTA_SSL.2, FTA_SSL.3, FTA_TSE.1, FTP_TRP.1
Account Logon	Audit when a DC receives a logon request.	FIA_SOS.1, FIA_UAU.1, FIA_UAU.6, FIA_UID.1, FTA_MCS.1

6.1.1.2 Audit Log Review

The Event Viewer administrator tool provides a user interface to view, sort, and search the security log. The security log can be sorted and searched by user identity, event type (by category and event ID), date, time, source, outcome (success and/or failure), and computer. The security log can also be searched by free form text occurring in the audit records. For example, this enables searching based on object identifiers.

6.1.1.3 Selective Audit

The authorized administrator is provided the ability to select events to be audited based upon object identity, user identity, workstation (host identity), type (category), and outcome (success or failure) of the event.

6.1.1.4 Audit Log Overflow Protection

The TSF protects against the loss of events through a combination of controls associated with audit queuing and event logging. As configured in the TOE, audit data is appended to the audit log until it is full. The TOE protects against lost audit data by allowing the authorized administrator to configure the system to generate an audit event when the security log reaches a specified capacity percentage (e.g., 90%). Additionally, the authorized administrator can configure the system not to overwrite events and to shutdown when the security log is full. When so configured, after the system has shutdown due to audit overflow, only the authorized administrator can log on. When the security log is full, a message is written to the terminal display of the authorized administrator indicating the audit log has overflowed.

As described earlier, the TSF collects audit data in two ways, via the SRM and via the LSA server. Both components maintain audit event queues. The SRM puts audit records on an internal queue to be sent to the LSA server. The LSA maintains a second queue where it holds the audit data from SRM and the other services in the security process. Both audit queues detect when an audit event loss has occurred. The SRM service maintains a high water mark and a low water mark on its audit queue to determine when full. The LSA also maintains marks in its queue to indicate when it is full.

Audit events may be lost if the SRM or the LSA queues reach their high-water mark, or if the security log file is full. The TOE can be configured to crash when the audit trail is full. The security log file is limited in size by the resources available on the system.

6.1.1.5 Audit Log Restricted Access Protection

The Event Logger controls and protects the security event log. Note that the underlying files are configured so that only the TSF can open the files and the Event Logger opens those files exclusively when it starts and keeps them open while it is running. To view the contents of the security log, the user must be an authorized administrator. The security event log is a system resource, created during system startup. No interfaces exist to create, destroy, or modify a security event within the security event log. The LSA subsystem is the only service registered to enter events into the security log. The TOE only offers user interfaces to read and clear the security event log and these interfaces require the user to be an authorized administrator.

SFR Mapping:

The **Audit function** satisfies the following SFRs:

- FAU_GEN.1: The TOE audit collection is capable of generating audit events for items identified in Table 6-1, TOE audit events. For each audit event the TSF records the date, time, user Security Identifier (SID) or name, logon type (for logon audit records), event ID, source, type, and category.
- FAU_GEN.2: All audit records include the user SID, which uniquely represents each user.
- FAU_SAR.1 – The event viewer provides authorized administrators with the ability to review audit data in a readable format.
- FAU_SAR.2 and FMT_MTD.1b: Only authorized administrators have any access to the audit log.
- FAU_SAR.3: The audit function provides capabilities for selective auditing and review using the event viewer. The TOE provides the capability to select events to be audited based on the success and/or failure at the category level. Additionally, for the object access category of events, events can be selected based on user identity. The TSF determines which audit events to record based on the current audit policy and the specific settings in the SACLs. The event viewer provides the capability to perform searches and sorting of audit data by date, time, user SID or name, computer, event ID, source, type, and category. Additionally, the event viewer provides the capability to perform searching based upon specified free form text substrings within the audit records (e.g., to search for specific object identifiers).
- FAU_SEL.1: The TSF provides the ability for the authorized administrator to select the events to be audited based upon object identity, user identity, workstation (host identity), event type, and success or failure of the event.
- FAU_STG.1: The interface to the security log is limited by the event logger. The interface to the security log only allows for viewing the audit data and for clearing all the audit data. The interface to the security log is restricted to authorized administrators and does not allow for the modification of audit data within the security log.
- FAU_STG.3: The authorized administrator can configure the system such that an audit event (alarm) is generated if the audit data exceeds a specified percentage of the security log.
- FMT_MTD.1a (partial): The TSF restricts the ability to specify the size of the security log to an authorized administrator.
- FAU_STG.4: The TOE can be configured such that when the security log is full the system shuts down. At that point, only the authorized administrator can log on to the system to clear the security log and return the system to an operational state consistent with TOE guidance. Additionally, when the security log reaches a certain percentage, an audit event (alarm) is generated.

6.1.2 User Data Protection Function

The User Data Protection security services provided by the TOE are:

- Discretionary Access Control
- Mandatory Integrity Control
- WEBUSER Access Control
- Content Provider Access Control
- Information Flow Control and Protection

- Residual Data Protection

6.1.2.1 Discretionary Access Control (DAC)

The TSF mediates access between subjects and user data objects, also known as named objects. Subjects consist of processes with one or more threads running on behalf of users. **Table 6-2** lists the specific user data objects under the control of the DAC policy for the TOE.

Table 6-2 Named Objects

Name	Description
Desktop	The primary object used for graphical displays.
Event	An object created for the interprocess communication mechanism.
Keyed Event	An object created for the interprocess communication mechanism.
Event Pair	An object created for the interprocess communication mechanism.
I/O Completion Port	An object that provides a means to synchronize I/O.
Job	An object that allows for the management of multiple processes as a unit.
Registry Key	Registry Keys are the objects that form the Registry.
Mutant	An object created for the interprocess communication mechanism (known as a mutex in the Windows API).
Object Directory	A directory in the object namespace.
ALPC Port	A connection-oriented local process communication mechanism object that supports client and server side communication end points such as message queues.
Mailslot	An I/O object that provides support for message passing IPC via the network.
Named Pipe	An I/O object used for IPC over the network.
NTFS Directory	NT file system file object.
NTFS File	A user data file object managed by NTFS.
Printer	Represents a particular print queue and its association with a print device.
Active Directory	Represents shared resources defined and maintained by Active Directory services. Objects in the Active Directory data store are also referred to as “Directory Store objects” (DS objects).
Process	An execution context for threads that has associated address space and memory, token, and handle tables.
Section	A memory region.
Semaphore	An object created for interprocess communication mechanism.
Symbolic Link	A means for providing name aliasing in the object name space.
Thread	An execution context (registers, stacks). All user-mode threads are associated with a process.
Timer	A means for a thread to wait for a specified amount of time to pass.
[Security] Token	This object represents the security context of a process or thread.
Volume	A partition or collection of partitions that have been formatted for use by a file system.
Window Station	A container for desktop objects and related attributes.
Application Pool File	A group of web applications that share configuration settings.

Name	Description
URL Reservation	A URL.
Debug	A set of resources used for debugging a process.
Filter Connection Port	Represents a mini-filter driver.
Filter Communication Port	Represents a port to communicate with a mini-filter driver.
[Transaction] Enlistment	An object representing a transactional enlistment. An enlistment is an association between a resource manager and a transaction.
Transaction	An object that defines a logical unit of work.
ResourceManager	An object used to manage the data that is associated with each transaction.
TransactionManager	An object used to track the state of each transaction and coordinates recovery operations after a system crash.

6.1.2.1.1 Subject DAC Attributes

Tokens contain the security attributes for a subject. Tokens are associated with processes and threads running on behalf of the user. The DAC related information in the token includes: the Security Identifier (SID) for the user, SIDs representing groups for which the user is a member, privileges assigned to the user, an owner SID that identifies the SID to assign as owner for newly created objects, a default Discretionary Access Control List (DACL) (for newly created objects), token type (primary or impersonation), the impersonation level (for impersonation tokens), an optional list of restricting SIDs, and a logon ID for the session.

As described in the I&A function, a thread can be assigned an impersonation token that would be used instead of the process' token when making access checks and generating audit data. Hence, that thread is impersonating the client that provided the impersonation token. Impersonation stops when the impersonation token is removed from the thread or when the thread terminates.

A token may also include a list of restricting SIDs which are used to limit access to objects. Restricting SIDs are contained in restricted tokens, (which is a special form of a thread impersonation token), and when configured serve to limit the corresponding process access to no more than that available to the restricted SID.

Access decisions are made using the impersonation token of a thread if it exists, and otherwise the thread's process primary token (which always exists).

6.1.2.1.2 Object DAC Attributes

Security Descriptors (SDs) contain all of the security attributes associated with an object. All objects in Table 6-2 have an associated SD. The security attributes from a SD used for access control are the object owner SID, the DACL present flag, and the DACL itself, if present.

DACLs contain a list of Access Control Entries (ACEs). Each ACE specifies an ACE type, a SID representing a user or group, and an access mask containing a set of access rights. Each ACE has inheritance attributes associated with it that specify if the ACE applies to the associated object only, to its children objects only, or to both its children objects and the associated object.

There are two types of ACEs that apply to access control:

- ALLOW ACES
    - ACCESS_ALLOWED_ACE: used to grant access to a user or group of users.
    - ACCESS_ALLOWED_OBJECT_ACE: (for DS objects) used to grant access for a user or group to a property or property set on the directory service object, or to limit the ACE_inheritance to a specified type of child object. This ACE type is only supported for directory service objects.
- DENY ACES
    - ACCESS_DENIED_ACE: used to deny access to a user or group of users.
    - ACCESS_DENIED_OBJECT_ACE: (for DS objects) used to deny access for a user or group to a property or property set on the directory service object or to limit the ACE_inheritance to a specified type of child object. This ACE type is only supported for directory service objects.

In the ACE, an access mask contains object access rights granted (or denied) to the SID, representing a user or group. An access mask is also used to specify the desired access to an object when accessing the object and to identify granted access associated with an opened object. Each bit in an access mask represents a particular access right. There are four categories of access rights: standard, specific, special, and generic. Standard access rights apply to all object types. Specific access rights have different semantic meanings depending on the type of object. Special access rights are used in desired access masks to request special access or to ask for all allowable rights. Generic access rights are convenient groupings of specific and standard access rights. Each object type provides its own mapping between generic access rights and the standard and specific access rights.

For most objects, a subject requests access to the object (e.g., opens it) and receives a pointer to a handle in return. The TSF associates a granted access mask with each opened handle. For kernel-mode objects, handles are maintained in a kernel-mode handle table. There is one handle table per process; each entry in the handle table identifies an opened object and the access rights granted to that object. For user-mode TSF servers, the handle is a server-controlled context pointer associated with the connection between the subject and the server. The server uses this context handle in the same manner as with the kernel mode (i.e., to locate an opened object and its associated granted access mask). In both cases (user and kernel-mode objects), the SRM makes all access control decisions.

For some objects (in particular, DS objects), the TSF does not maintain an opened context (e.g., a handle) to the object. In these cases, access checks are performed on every reference to the object (in place of checking a handle’s granted access mask). DS objects also differ from other objects in that they have additional attributes, known as properties and property sets (groups of properties). Properties reference specific portions of a DS object. Property sets reference a collection of properties. Every DS object, property set and property has an associated object type GUID. The TOE allows access control for

DS objects to the level of GUIDs (i.e., the entire DS object, a given property set, and or a specific property). Like all objects, DS objects still have a single security descriptor for the entire object; however the DACL for a DS object can contain ACEs the grants/denies access to any of the associated GUIDs.

6.1.2.1.3 DAC Enforcement Algorithm

The TSF enforces the DAC policy to objects based on SIDs and privileges in the requestor's token, the desired access mask requested, and the object's security descriptor.

Below is a summary of the algorithm used to determine whether a request to access a user data object is allowed. In order for access to be granted, all access rights specified in the desired access mask must be granted by one of the following steps. At the end of any step, if all of the requested access rights have been granted then access is allowed. At the end of the algorithm, if any requested access right has not been granted, then access is denied.

1. Privilege Check:
    a. Check for SeSecurity privilege: This is required if ACCESS_SYSTEM_SECURITY is in the desired access mask. If ACCESS_SYSTEM_SECURITY is requested and the requestor does not have this privilege, access is denied. Otherwise ACCESS_SYSTEM_SECURITY is granted.
    b. Check for SeTakeOwner privilege: If the desired mask has WRITE_OWNER access right, and the privilege is found in the requestor's token, then WRITE_OWNER access is granted.
    c. Check for SeBackupPrivilege: The Backup Files and Directories privilege allows a subject to read files and registry objects for backup operations regardless of permissions. If the privilege is held and the operation is a backup operation, no further checking is performed and access is allowed. Otherwise this check is irrelevant and the access check proceeds.
    d. Check for SeRestorePrivilege: The Restore Files and Directories privilege allows a subject to write files and registry objects for restore operations regardless of permissions. If the privilege is held and the operation is a restore operation no further checking is performed, and access is allowed. Otherwise this check is irrelevant and the access check proceeds.
    e. Check for SeSyncAgentPrivilege: The Synchronize Directory Service data privilege allows a subject to read Active Directory objects for synchronization operations regardless of permissions. If the privilege is held and the operation is a synchronization operation no further checking is performed, and access is allowed. Otherwise this check is irrelevant and the access check proceeds.
2. Owner Check:
    a. If the DACL contains one or more ACEs with the OwnerRights SID, those entries, along with all other applicable ACEs for the user, are used to determine the owner's rights.

   b. Otherwise, check all the SIDs in the token to determine if there is a match with the object owner. If so, the READ_CONTROL and WRITE_DAC rights are granted if requested.
3. DACL not present:
   a. All further access rights requested are granted.
4. DACL present but empty:
   a. If any additional access rights are requested, access is denied.
5. Iteratively process each ACE in the order that they appear in the DACL as described below:
   a. If the inheritance attributes of the ACE indicate the ACE is applicable only to children objects of the associated object, the ACE is skipped.
   b. If the SID in the ACE does not match any SID in the requestor's access token, the ACE is skipped.
   c. If a SID match is found, and the access mask in the ACE matches an access in the desired access mask:
      i. Access Allowed ACE Types: If the ACE is of type ACCESS_ALLOWED_OBJECT_ACE and the ACE includes a GUID representing a property set or property associated with the object, then the access is granted to the property set or specific property represented by the GUID (rather than to the entire object). Otherwise the ACE grants access to the entire object.
      ii. Access Denied ACE Type: If the ACE is of type ACCESS_DENIED_OBJECT_ACE and the ACE includes a GUID representing a property set or property associated with the object, then the access is denied to the property set or specific property represented by the GUID. Otherwise the ACE denies access to the entire object. If a requested access is specifically denied by an ACE, then the entire access request fails.
6. If all accesses are granted but the requestor's token has at least one restricting SID, the complete access check is performed against the restricting SIDs. If this second access check does not grant the desired access, then the entire access request fails.

6.1.2.1.4 DAC Enforcement of Encrypted Files

The TOE provides the ability to encrypt NTFS file objects. Users may encrypt files at their discretion. If a file is encrypted, the TSF performs checks in addition to the checks presented in the DAC Enforcement Algorithm upon subsequent access request to the encrypted file.

The first time a user encrypts a file the TSF assigns the user account a public/private key pair. Every time a user encrypts a file, the TSF creates a randomly generated File Encryption Key (FEK) to protect the file. The FEK is used to encrypt the file data using (by default) the AES-256 algorithm. The TSF stores the FEK as an attribute of the file and encrypts the FEK using the RSA public-key based encryption algorithm associated with the user's public key. The TSF also allows a user who can decrypt the file to grant [EFS] access to other users by adding additional encrypted FEKs (encrypted with the shared users' public key) to the file. An authorized administrator can assign a public/private key pair to any number of accounts. These accounts are referred to as recovery agents and the TSF encrypts the FEK with the public key of the recovery agent. The purpose of recovery keys is to let designated accounts, or Recovery Agents, decrypt a user's file when administrative authority must have access to the user's data.

After a file is encrypted, upon subsequent access request, the TSF checks that the user private key or recovery private key can decrypt the encrypted FEK. There may be more than one encrypted FEK associated with the file. In this case, the TSF attempts to decrypt each associated encrypted FEK (each of which is encrypted) until it is successfully decrypted or it reaches the end of the list of FEKs.

If the FEK is decrypted successfully with the private key, the decrypted FEK is then used to decrypt the file contents and the access request is granted. If the TSF cannot decrypt any of the encrypted FEKs associated with the file using the user private key or the recovery key, the access request is not granted.

EFS allows users to export applicable FEKs to smart cards so that the encrypted file could be accessed from another instance of the TOE using the FEK on the smart card. Storing the FEK only on a smart card also offers users more direct control of the FEK which could offer additional security for some applications. Additionally, EFS has been revised to support encryption of the paging file so that there is less risk of sensitive data disclosure should the page file remain on the system volume while the TOE is not in operation.

6.1.2.1.5 Default DAC Protection

The TSF provides a process ensuring a DACL is applied by default to all new objects. When new objects are created, the appropriate DACL is determined. The default DAC protections for DS object and that for non-DS objects are slightly different.

The TOE uses the following rules to set the DACL in the SDs for new non-DS securable objects:

- The object's DACL is the DACL from the SD specified by the creating process. The TOE merges any inheritable ACEs into the DACL unless SE_DACL_PROTECTED is set in the SD control flags. The TOE then sets the SE_DACL_PRESENT SD control flag. Note that a creating process can explicitly provide a SD that includes no DACL. The result will be an object with no protections. This is distinct from providing no SD which is described below.
- If the creating process does not specify a SD, the TOE builds the object's DACL from inheritable ACEs in the parent object's DACL. The TOE then sets the SE_DACL_PRESENT SD control flag.
- If the parent object has no inheritable ACEs, the TOE uses its object manager subcomponent to provide a default DACL. The TOE then sets the SE_DACL_PRESENT and SE_DACL_DEFAULTED SD control flags.
- If the object manager does not provide a default DACL, the TOE uses the default DACL in the subject's access token. The TOE then sets the SE_DACL_PRESENT and SE_DACL_DEFAULTED SD control flags.
- The subject's access token always has a default DACL, which is set by the LSA subcomponent when the token is created.

The method used to build a DACL for a new DS object is slightly different. There are two key differences, which are as follows:

- The rules for creating a DACL distinguish between generic inheritable ACEs and object-specific inheritable ACEs in the parent object's SD. Generic inheritable ACEs can be inherited by all types of child objects. Object-specific inheritable ACEs can be inherited only by the type of child object to which they apply.
- The AD schema for the object can include a SD. Each object class defined in the schema has a defaultSecurityDescriptor attribute. If neither the creating process nor inheritance from the parent object provides a DACL for a new AD object, the TOE uses the DACL in the default SD specified by the schema.

The TOE uses the following rules to set the DACL in the security descriptor for new DS objects:

- The object's DACL is the DACL from the SD specified by the creating process. The TOE merges any inheritable ACEs into the DACL unless SE_DACL_PROTECTED is set in the SD control flags. The TOE then sets the SE_DACL_PRESENT SD control flag.
- If the creating process does not specify a SD, the TOE checks the parent object's DACL for inheritable object-specific ACEs that apply to the type of object being created. If the parent object has inheritable object-specific ACEs for the object type, the TOE builds the object's DACL from inheritable ACEs, including both generic and object-specific ACEs. It then sets the SE_DACL_PRESENT SD control flag.
- If the parent object has no inheritable object-specific ACEs for the type of object being created, the TOE uses the default DACL from the AD schema for that object type. It then sets the SE_DACL_PRESENT and SE_DACL_DEFAULTED SD control flags.
- If the AD schema does not specify a default DACL for the object type, the TOE uses the default DACL in the subject's access token. It then sets the SE_DACL_PRESENT and SE_DACL_DEFAULTED SD control flags.
- The subject's access token always has a default DACL, which is set by the LSA subcomponent when the token is created.

All tokens are created with an appropriate default DACL, which can be applied to the new objects as appropriate. The default DACL is restrictive in that it only allows the SYSTEM SID and the user SID that created the object to have access. The SYSTEM SID is a special SID representing TSF trusted processes.

6.1.2.1.6 Reference Mediation

Access to objects on the system is generally predicated on obtaining a handle to the object. Handles are usually obtained as the result of opening or creating an object. In these cases, the TSF ensures that access validation occurs before creating a new handle for a subject. Handles may also be inherited from a parent process or directly copied (with appropriate access) from another subject. In all cases, before creating a handle, the TSF ensures that that the security policy allows the subject to have the handle (and thereby access) to the object. A handle always has a granted access mask associated with it. This mask indicates, based on the security policy, which access rights to the object that the subject was granted. On every attempt to use a handle, the TSF ensures that the action requested is allowed according to the handle’s granted access mask. In a few cases, such as with DS, objects are directly accessed by name without the intermediate step of obtaining a handle first. In these cases, the TSF checks the request against the access policy directly (rather than checking for a granted access mask).

6.1.2.2 Mandatory Integrity Control

In addition to discretionary access control, the TSF provides mandatory integrity control (MIC). MIC uses integrity levels and mandatory policies to evaluate access. Processes (i.e., subjects) and most securable objects (see **Mandatory Integrity Control Policy (FDP_ACC.2d))** for the applicable list of objects) are assigned integrity levels that determine their levels of protection or access. For example, a subject with a low integrity level cannot write to an object with a medium integrity level, even when that object's DACL allows write access to the subject.

Integrity labels specify the integrity levels of securable objects and processes. Integrity labels are represented by integrity SIDs. The integrity SID for a securable object is stored in its SACL. The SACL contains a SYSTEM_MANDATORY_LABEL_ACE ACE that in turn contains the integrity SID. Any object without an integrity SID is treated as if it had medium integrity. The integrity SID for a process is stored in its access token.

The integrity labels defined in Windows are:

- **Untrusted**: Used by processes started by the Anonymous group.
- **Low**: Used by protected mode (specifically for Internet Explorer), blocks write access to most objects (such as files and registry keys) on the system.
- **Medium:** Normal applications being launched while user account control (UAC) is enabled.
- **High:** Applications launched through administrator elevation when UAC is enabled, or normal applications if UAC is disabled.
- **System**: Services and other system-level applications (such as WinLogon).

Each process has a mandatory policy represented by its TOKEN_MANDATORY_POLICY which can have one of the following values:

- TOKEN_MANDATORY_POLICY_OFF: No mandatory policy is enforced for the access token.
- TOKEN_MANDATORY_POLICY_NO_WRITE_UP: The mandatory policy is enforced and the subject cannot write objects with higher integrity labels.
- TOKEN_MANDATORY_POLICY_NEW_PROCESS_MIN: A process that is created is assigned an integrity label that is the lesser of the parent-process and that of the executable file for the process.
- TOKEN_MANDATORY_POLICY_VALID_MASK: A combination of TOKEN_MANDATORY_POLICY_NO_WRITE_UP and TOKEN_MANDATORY_POLICY_NEW_PROCESS_MIN.

By default processes are assigned TOKEN_MANDATORY_POLICY_VALID_MASK.

Processes can access objects that have an integrity level lower than or equal to their own integrity level. The SYSTEM_MANDATORY_LABEL_ACE ACE in the SACL of a securable object contains an access mask that specifies the access that subjects with integrity levels lower than the object are granted (i.e., the mandatory policy for the object). The values defined for this access mask are:

- SYSTEM_MANDATORY_LABEL_NO_WRITE_UP: A subject with a lower integrity label cannot write an object with a higher integrity label.
- SYSTEM_MANDATORY_LABEL_NO_READ_UP: A subject with a lower integrity label cannot read an object with a higher integrity label.
- SYSTEM_MANDATORY_LABEL_NO_EXECUTE_UP: A subject with a lower integrity label cannot execute an object with a higher integrity label.

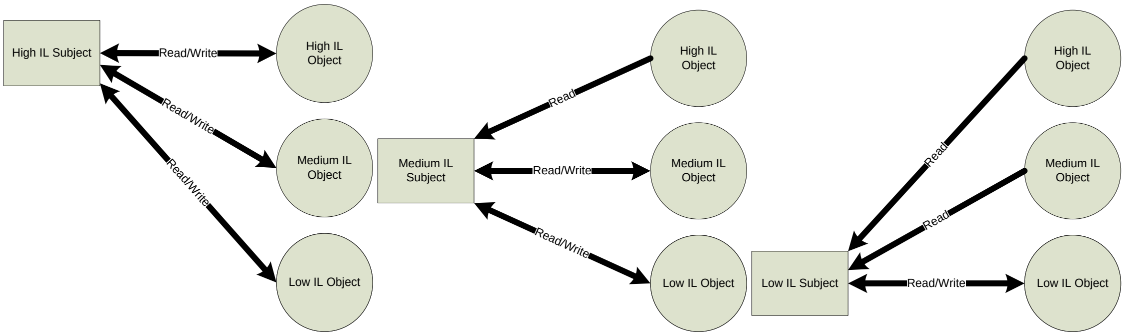
By default, every object, except processes and threads, has an access mask of SYSTEM_MANDATORY_LABEL_NO_EXECUTE_UP. Processes and threads have an access mask of SYSTEM_MANDATORY_LABEL_NO_READ_UP.

Note that both the process policy and the object policy are applied simultaneously whenever a subject attempts to access an object. The allowed access will effectively be the logical intersection of the respective policies. However, if a process does not have the TOKEN_MANDATORY_POLICY_NO_WRITE_UP value (i.e., either TOKEN_MANDATORY_POLICY_OFF or TOKEN_MANDATORY_POLICY_NEW_PROCESS_MIN, then the object label and policy are irrelevant.

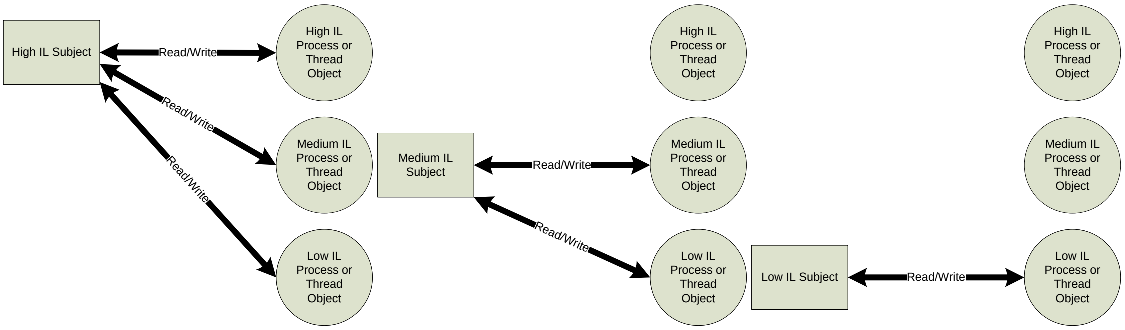
In the default cases, the MIC policy rules are twofold:

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) or execute (when applicable for the object) is permitted.
2. If the integrity label of the object is less 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.

The rules for hierarchical integrity attribute schemes as defined by the MIC rules above are reflected in the following three diagrams.



By default, process and thread objects are an exception to the integrity policy rules implemented by Windows. For these objects there is a stipulation of “no read up”. This is reflected in the following three diagrams.


High IL Subject
Read/Write
Read/Write
Read/Write
High IL Process or Thread Object
Medium IL Process or Thread Object
Low IL Process or Thread Object
Medium IL Subject
Read/Write
Read/Write
High IL Process or Thread Object
Medium IL Process or Thread Object
Low IL Process or Thread Object
Low IL Subject
Read/Write
High IL Process or Thread Object
Medium IL Process or Thread Object
Low IL Process or Thread Object

When an object is created, it is assigned an integrity label equal to that of the creating process. Subsequently, only a process with the "modify an object label" privilege (i.e., SeRelabelPrivilege, assigned to an authorized administrator) can change the label of the object.

Processes associated with non-administrative users receive a medium integrity level by default (e.g., when they log in). Processes associated with administrative users receive a high integrity level by default. Processes started by another subject are assigned the lower of the integrity level assigned to the subject or the integrity level assigned to the executable file associated with the subject, unless the mandatory policy for the process does not indicate TOKEN_MANDATORY_POLICY_NEW_PROCESS_MIN in which case the integrity label of the executable file will be assigned.

6.1.2.3 WEBUSER Access Control

The TOE includes a web server (the IIS) in the Windows 2008 server product that mediates access requests to its web server content from clients accessing the web server through HTTP.

IIS supports user authentication using either anonymous, basic, digest, certificate, NT or Windows Live ID (i.e., an Internet accessible authentication scheme outside the control of the TOE) authentication scheme. In an evaluated configuration, an IIS server accepts only the anonymous, digest, certificate, and NT authentication schemes. Thus, only HTTP requests from clients that authenticate using an acceptable scheme are processed by the web server. Note that IIS anonymous authentication allows a web server request to be serviced without prompting the client for I&A. However, that client has been authenticated prior to making a web server request in the evaluated configuration. The web server then assigns the connection to the user account that is specified for anonymous connections.

IIS ensures that the DAC Policy of the files associated with the web server content requested is enforced. Therefore, the DACL of the file associated with the web content is compared against the user ID and group ids associated with the web user requesting the web content. If a request to access web content from a web user is other than a request to read web content the request is denied unless certain configurations of web permissions are associated with that web content.

In addition to ensuring that the DAC policy is enforced, IIS enforces further restrictions to web content based upon web permissions that are associated with web content in IIS configuration repository, referred to as the metabase. Web permissions do not violate the DAC policy and access can only be further restricted by IIS.

IIS allows for configuration settings to be associated with a URL that associate web permissions with URLs (i.e., URL authorization). If configured, these settings allow for access control checks to be performed by IIS when access request are made to these URLs. These web permissions control the ability to perform the following actions to web content:

- Access URL: access the URL
- Read web permission: read web content
- Write web permission: change web content
- Execute web permission: execute web content
- Source web permission: view the source of web content
- Browsing web permission: view the file lists and collections in a directory

If web content is configured with web permissions, then IIS performs additional checks when an access request is made for that web content to ensure that the appropriate permission is configured for that web content (as described above). If the requested permission is not one of those defined above, access will be denied. If the appropriate permission is configured, access will be granted. For example, if a *write* request is made to web content and that web content is not configured with the *write* web permission then the request will be denied. However, if the *write* request is made to web content and the DACL associated with the file allows write access to that user and the *write* web permission is configured for that web content, then access is granted.

Under certain circumstances IIS denies access to web content based upon web permissions associated with the web content, as follows:

- If web content is configured to require SSL/TLS and the web user request access via HTTP and not Secure HTTP (HTTPS), then access is denied.
- If web content is configured to require SSL/TLS and use a client certificate, and the web user request access via HTTPS without a certificate or via HTTP, then access is denied
- If web content is configured to require SSL/TLS and a negotiated certificate or requires a certificate, and the web user request access via HTTP or via HTTPS with an invalid or revoked certificate, then access is denied.
- If the authorization setting of a web user requesting access does not match the configured authorization setting associated with the web content, then access is denied.
- If the client certificate mapping setting of the web user requesting access does not match the configured certificate mapping setting associated with the web content, then access is denied.

In the evaluated configuration execute permission of web content is not allowed.

Read access to web content is allowed by default, however, other access must be specifically assigned by the authorized administrator.

6.1.2.3.1 WEBUSER Data Integrity and Confidentiality

When configured to do so, IIS protects data during transmission between the web user and the web server from unauthorized disclosure and modification by requiring that the web user must use HTTPS with or without a client certificate which is accomplished by configuring the web content object to require SSL/TLS. Additionally, by requiring SSL/TLS, IIS can determine upon receipt of data from the web user if data content has been modified.

6.1.2.4 Content Provider Access Control

A web user that is allowed to install and modify web content is referred to as a content provider. The IIS configuration values that define the configuration of web permissions to web content objects are stored in what is referred to as a metabase file. This metabase can only be manipulated by authorized administrators. Access request to modify web content are mediated based upon the same rules as described for web users.

6.1.2.5 Information Flow Control and Protection

The TOE includes a set of Windows 7 and Windows Server 2008 R2 systems that can be connected via their network interfaces. Each Windows 7 and Windows Server 2008 R2 system within the TOE provides a subset of the TSFs. Therefore, the TSF for Windows 7 and Windows Server 2008 R2 can be a collection of SFs from an entire network of systems (in the case of domain configurations). Therefore, the TSF is considered to be the collection of the TSFs of each Windows 7 and Windows Server 2008 R2 system included in the TOE.

The TOE uses a suite of Internet standard protocols including IPSec and ISAKMP. IPSec can be used to secure traffic using IP addresses or port number between two computers or between two TSFs within the TOE.

IPSec policies specify the functions that IPSec must perform for a given outbound or inbound packet and include a list of filters to be applied to IP packet traffic. Filters can be specified to control traffic flow based upon source IP address, destination IP address, protocol, source port, or destination port. An action of permit or block can be specified within the filter for specific flows of traffic based upon source IP address, destination IP address, protocol, source port, or destination port.

The TSF enforces these filters before sending any outbound packets and before allowing any inbound packets to proceed.

The TSF also prevents the disclosure and modification of user data using IPSec policies and filters. IPSec policies and filters can be configured only by an authorized administrator and can be configured to apply actions to specify traffic flow characteristics such as encrypting or signing. IPSec uses the CNG algorithms to provide data confidentiality and integrity for IP packets.

See Section 6.1.6.2, Internal TOE Protection, for further details of IPSec.

The TSF allows for the authorized administrator to define a Connection Firewall policy that can specify what ports the TSF will allow connections upon. Using IPSec, this policy will then enforce the blocking of all other incoming connections and allows in only that which is a reply to a previous request that went out.

If the Windows Firewall feature is enabled by the authorized administrator, the TSF enforces the Connection Firewall policy that will block all unsolicited incoming packets except for packets destined for ports specified by the authorized administrator. To support this policy the TSF uses TCP/IP (IPv4 or IPv6).

When Windows Firewall is enabled, it opens and closes the communications ports that are used by authorized applications. Windows Firewall maintains a table of connections that are initiated on behalf of the other systems on the "protected" side of the local network, and inbound Internet traffic can reach the "protected" network only when the table holds a matching entry. The administrator configures which "services" will be permitted by Windows Firewall. The administrator also configures Internet Control Message Protocol (ICMP) message handling. Service settings and ICMP options are per interface. Windows Firewall supports Stateful Packet Filtering and Port Mapping.

Note that the Windows Firewall is enabled by default on all products. However, while on Windows 7 the settings are configured to be restrictive to prevent unsolicited incoming requests, Windows Server 2008 R2 has a more permissive default policy so that clients (e.g., other Windows products) can access available services.

6.1.2.6 Residual Data Protection Function

The TOE ensures that any previous information content is unavailable upon allocation to subjects and objects. The TSF ensures that resources exported to user-mode processes do not have residual information in the following ways:

- All objects are based on memory and disk storage. Memory allocated for objects is either overwritten with all zeros or overwritten with the provided data before being assigned to an object. Objects stored on disk are restricted to only disk space used for that object. Read/write pointers prevent reading beyond the space used by the object. Only the exact value of what is most recently written can be read and no more. For varying length objects, subsequent reads only return the exact value that was set, even though the actual allocated size of the object may be greater than this.
- Subjects have associated memory and an execution context. The TSF ensures that the memory associated with subjects is either overwritten with all zeros or overwritten with user data before allocation as described in the previous bullet for memory allocated to objects. In addition, the execution context (registers) is initialized when new threads within a process are created and restored when a thread context switch occurs.

SFR Mapping:

The **User Data Protection** function satisfies the following SFRs:

- FDP_ACC.2a: The SRM mediates all access to objects, including kernel-based objects and user-mode TSF server-based objects. All access to objects is predicated on the SRM validating the access request. In the case of most objects, this DAC validation is performed on initial access (e.g., "open") and subsequent use of the object is via a handle that includes a granted access mask. For some objects (in particular DS objects), every reference to the object requires a complete DAC validation to be performed. The TSF mediates read access by subjects to encrypted files by protecting user and recovery private keys and using those keys to protect the FEK.

- FDP_ACF.1a: The TSF enforces access to user objects based on SIDs and privileges associated with subjects contained in tokens (impersonation token, if one exist), and the security descriptors for objects. The rules governing access are defined as part of the DAC algorithm described above. The TSF uses the FEKs associated with the file and protected using authorized users' private keys to protect the encrypted file contents.
- FDP_ACC.2b, FDP_ACC.2c, FDP_ACF.1b, and FDP_ACF.1c: The TSF enforces access to web server content based upon the web user's identity and group memberships, the DACL associated with the object, URL authorization, and web permissions. The WEBUSER policy rules govern access to read the web content and modify the web content if specifically authorized (FDP_ACC.2b, FDP_ACF.1b). The CONTENT PROVIDER policy rules govern access to primarily control the ability to make web content available to web users and to modify web content (FDP_ACC.2c, FDP_ACF.1c).
- FDP_ACC.2d and FDP_ACF.1d: The TSF enforces a Mandatory Integrity Control policy for process access to most objects covered by the DAC policy. The rules are enforced to ensure that process accesses to objects conform to rules that involve applicable attributes on the processes and objects as summarized earlier.
- FDP_IFC.1a, FDP_IFF.1a: The TSF controls the flow of traffic from one Windows 7 and Windows Server 2008 R2 system's TSF to another using the IPSec's capability to enforce filters that can be configured to restrict the flow of traffic based upon source IP address, destination IP address, source port, destination port, and protocol.
- FDP_IFC.1b, FDP_IFF.1b: The TSF controls the flow of traffic into a Windows 7 and Windows Server 2008 R2 system's TSF by providing the capability to block all unsolicited traffic with the exceptions of traffic targeted to ports specified by the authorized administrator.
- FDP_UCT.1, FDP_UIT.1: The TSF protects data during transmission between the web user and the web server from unauthorized disclosure and modification by requiring that SSL/TLS is used to support this communication.
- FDP_ITT.1: The TSF prevents the disclosure and modification of user data using IPSec encryption and digital signature capabilities when user data is transmitted between different system
- FDP_RIP.2 - The TSF ensures that previous information contents of resources used for new objects are not discernable in the new object via zeroing or overwriting of memory and tracking read/write pointers for disk storage. Every process is allocated new memory and an execution context. Memory is zeroed or overwritten before allocation.
- FMT.MSA.1a, FMT_MSA.1b: The ability to change the DAC policy is controlled by the ability to change an object's DACL. The following are the four methods that DACL changes are controlled:
  - Object owner: Has implicit WRITE_DAC access.
  - Explicit DACL change access: A user granted explicit WRITE_DAC access on the DACL can change the DACL.

  - Take owner access: A user granted explicit WRITE_OWNER access on the DACL can take ownership of the object and then use the owner's implicit WRITE_DAC access.
  - Take owner privilege: A user with SeTakeOwner privilege can take ownership of the object and then user the owner's implicit WRITE_DAC access.
- FMT_MSA.1b: The TSF associates private keys with users. Only the owner of the private key used to protect the FEK associated with the file or an administrator or subject with a specific privilege can delete the FEK.
- FMT_MSA.1c: The ability to change the security attributes upon which the IPSec Filter Policy is based upon is restricted to the authorized administrator.
- FMT_MSA.1d: The ability to change the security attributes upon which the Connection Firewall Policy is based upon is restricted to the authorized administrator.
- FMT_MSA.1e, FMT_MSA1f: The ability to change the security attributes upon which the WEBUSER and CONTENT PROVIDER policies are based upon is restricted to the authorized administrator.
- FMT_MSA.1g: The ability to change Mandatory Integrity Control related security attributes is restricted to processes holding a specific privilege (i.e., SeRelabelPrivilege) allowing the modification of object labels.
- FMT.MSA.3a - The TSF provides restrictive default values for security attributes used to provide access control via the process's default DACLs which only allows access to the SYSTEM and the user creating the object. Users who create objects can specify a SD with a DACL to override the default. The initial keys are cryptographically generated and cannot be modified.
- FMT_MSA.3b: Filters can be defined and assigned to restrict traffic flow from one TSF to another. However, by default, there are no filters assigned and traffic is allowed to flow in an unrestricted manner. Only the authorized administrator can define or modify the IPSec filters that specify the rules for traffic flow.
- FMT_MSA.3c: By default, Windows 7 has a very restrictive default firewall policy while Server 2008 R2 has a permissive policy so that it can support client access to its services. Only the authorized administrator can specify ports for which unsolicited traffic will be accepted. However, the firewall feature is optional and can be disabled in the evaluated configuration in which case no restriction on traffic flow is enforced.
- FMT_MSA.3d, FMT_MSA.3e: By default, only read access to web content is allowed and only an authorized administrator can define the configuration or the web permissions associated with the web content in the metabase.
- FMT_MSA.3f: By default, objects and processes are assigned Mandatory Integrity labels and policies that prevent writing to higher integrity labels and read access to processes and threads at higher integrity labels. The defaults cannot be changed during process or object creation, though some attributes can be changed later per FMT_MSA.1(g).
- FMT_MTD.1a: Only an authorized administrator can modify the values in the metabase which include the IIS configuration. These values define permissions to web content.

- FMT_REV.1b: The ability to revoke access to an object is controlled by the ability to change the DACL and is governed by the same conditions for FMT_MSA.1a above. The changed DACL is effective upon subsequent access checks against the object.

6.1.3 Cryptographic Protection

Cryptography API: Next Generation (CNG) API is the long-term replacement for the CryptoAPI. CNG is designed to be extensible at many levels and cryptography agnostic in behavior. An important feature of CNG is its support for the Suite B algorithms. CNG includes support for Suite B that extends to all required algorithms: AES (all key sizes), the SHA-2 family (SHA-256, SHA-384 and SHA-512) of hashing algorithms, elliptic curve Diffie Hellman (ECDH), and elliptical curve DSA (ECDSA) over the NIST-standard prime curves P-256, P-384, and P-521.

Protocols such as the Internet Key Exchange (IKE, mainly used in IPsec), and Transport Layer Security (TLS), make use of elliptic curve Diffie-Hellman (ECDH) included in Suite B.

Random number generation (RNG) is provided in Suite B and is implemented in accordance with NIST Special Publication 800-90. Note that while the TOE can generate random numbers using only software-based noise sources, it will also utilize the hardware-based noise source available from a TPM chip when one is present. The random number generator is seeded by independent software-based entropy sources, and independent hardware-based entropy sources when a TPM chip is present. CNG components such as Asymmetric Key Generation, Signing, and the Schannel Protocol Provider use this RNG. The TSF defends against tampering of the random number generation (RNG)/ pseudorandom number generation (PRNG) sources by encapsulating its use of Suite B in Kernel Security Device Driver.

The encryption and decryption operations are performed by independent modules, known as Cryptographic Service Providers (CSPs). The CSPs, specifically the Cryptographic Primitives Library and kernel security device driver, are FIPS 140-2 Level 1 compliant. The TSF applies validation techniques to generate symmetric keys in accordance with NIST Special Publication 800-57, “Recommendation for Key Management.”

In addition to encryption and decryption services, the TSF provides other cryptographic operations such as hashing, key agreement, and digital signatures. The TSF also provides pseudo random number generation capabilities. These cryptographic capabilities are designed to conform to published standard and compliance with these cryptographic standards has been demonstrated as follows:

Table 6-3 Cryptographic Standards and Evaluation Methods

Cryptographic Operation	Standard	Evaluation Method
Encryption/Decryption	FIPS 46-3 - 3DES (aka TDEA) –CBC, ECB, and CFB	NIST CAVP #846 for TECB(e/d; KO 1,2), TCBC(e/d; KO 1,2), TCFB8(e/d; KO 1,2)
Encryption/Decryption	FIPS 197 - AES – ECB, CBC, CFB8, CCM, and GCM	NIST CAVP #1168 for ECB(e/d; 128,192,256), CBC(e/d; 128,192,256), CFB8(e/d; 128,192,256); #1178 and

Cryptographic Operation	Standard	Evaluation Method
		#1187 for CCM (KS: 128 , 192 , 256); and #1177 for CCM (KS: 128, 256)
Digital signature	FIPS 186-2 DSA	NIST CAVP #385, #386, #390, and #391 for KEYGEN(Y) MOD(1024), SIG(gen) MOD(1024), SIG(ver) MOD(1024)
Digital signature	rDSA	NIST CAVP #557 and #568 for ALG[RSASSA-PKCS1_V1_5] SIG(gen), SIG(ver), 1024 , 1536, 2048, 3072, 4096; #560 and #567 for ALG[RSASSA-PKCS1_V1_5] SIG(gen), SIG(ver), 1024 , 1536, 2048, 3072, 4096 and ALG[RSASSA-PSS] SIG(gen), SIG(ver), 1024 , 1536, 2048, 3072, 4096
Digital signature	ECDSA	NIST CAVP #141 and #142 for PKG: CURVES(P-256, P-384, P-521); SIG(gen): CURVES(P-256, P-384, P-521); and SIG(ver): CURVES(P-256, P-384, P-521)
Hashing	SHA-256, SHA-384, and SHA-512	NIST CAVP #1081 for SHA-256 (BYTE-only); SHA-384 (BYTE-only); and SHA-512 (BYTE-only)
Keyed-Hash Message Authentication Code	HMAC	NIST CAVP #673, #677, #686, and #687 for HMAC-SHA1; HMAC-SHA256; HMAC-SHA384; HMAC-SHA512; and #675 for HMAC-SHA1; HMAC-SHA256
Random number generation	NIST SP 800-90	NIST CAVP #27 and #24 for Dual_EC_DRBG and #23 for (No_df): AES-256
Random number generation	FIPS 186-2	NIST CAVP #649 for FIPS 186-2 [(x-Change Notice); (SHA-1)] FIPS 186-2 General Purpose [(x-Change Notice); (SHA-1)]
Key agreement	ECDSA (ANSI X9.62-1998)	Vendor Affirmed
Key agreement	ECDH (eliptic curve Diffie-Hellman) – NIST SP 800-56A	Vendor Affirmed
Key Generation	RNG (3DES and AES)	The random number was generated using NIST approved random number generators, in particular a NIST SP 800-90 DRNG (certificate #23)
Key Generation	RNG (DSA, rDSA, ECDSA, ECDH)	NIST CAVP #385, #386, #390, and #391 for KEYGEN(Y) MOD(1024); #559 for ALG[ANSIX9.31] Key(gen)(MOD: 1024, 1536, 2048, 3072, 4096; PubKey Values:

Cryptographic Operation	Standard	Evaluation Method
		65537); and #141 and #142 for PKG: CURVES(P-256, P-384, P-521)
Key Zeroization	FIPS 140-2	FIPS 140-2 certificates #1319, #1321, #1326, #1327, #1328, #1329, #1330, #1331, #1332, #1333, #1334, #1335, #1336, #1337, #1338, and #1339

The TSF includes a key isolation service designed specifically to host secret and private keys in a protected process to mitigate tampering or access to sensitive key materials. The TSF performs key entry and output in accordance with FIPS 140-2. The TSF performs a key error detection check on each transfer of key (internal, intermediate transfers). The TSF prevents archiving of expired (private) signature keys. The TSF destroys non-persistent cryptographic keys – note that all keys subject to destruction are stored within the cryptomodule that was subject to FIPS 140-2 certification - after a cryptographic administrator-defined period of time of inactivity. The TSF overwrites each intermediate storage area for plaintext key/critical cryptographic security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data). This overwriting is performed as follows:

- For non-volatile memories other than EEPROM and Flash, the overwrite is executed three or more times using a different alternating data pattern each time upon the transfer of the key/critical cryptographic security parameter to another location.
- For volatile memory and non-volatile EEPROM and Flash memories, the overwrite is a single direct overwrite consisting of a pseudo random pattern, followed by a read-verify upon the transfer of the key/critical cryptographic security parameter to another location.

SFR Mapping:

The **Cryptographic Protection** function satisfies the following SFRs:

- FCS_BCM_EXT.1: See **Table 6-3 Cryptographic Standards and Evaluation Methods**.
- FCS_COP.1a, FCS_COP.1d: The TSF uses the 3DES or AES (128-bit and higher key sizes) algorithm to encrypt user data and only allows the user who encrypted the data to decrypt the data by ensuring that the SID of the subject requesting decryption is the same as the SID of the subject that requested encryption of the data.
- FCS_COP.1a-g: See **Table 6-3 Cryptographic Standards and Evaluation Methods**.
- FCS_CKM.1a-b: See **Table 6-3 Cryptographic Standards and Evaluation Methods**.
- FCS_CKM.4: See **Table 6-3 Cryptographic Standards and Evaluation Methods**.
- FCS_COA_EXT.1: See **Table 6-3 Cryptographic Standards and Evaluation Methods.**
- FCS_RBG_EXT.1: See **Table 6-3 Cryptographic Standards and Evaluation Methods**.

6.1.4 Identification and Authentication Function

The TOE requires each user to be identified and authenticated prior to performing TSF-mediated functions on behalf of that user, with a few exceptions, regardless of whether the user is logging on interactively or is accessing the system via a network connection. One exception is the function allowing a user to shut the system down; however, an authorized administrator may disable even that function if it is not appropriate for a given environment. The other exception is access to the web server when anonymous authentication is allowed (as described in the **WEBUSER (WU) Complete Access Control (FDP_ACC.2b)** section) during which a HTTP request is serviced by the IIS web server without prompting the client for identification and authentication, even though that client may have been authenticated prior to making a web server request in the evaluated configuration.

6.1.4.1 Logon Type

Windows supports the following types of user logon:

Table 6-4 Logon Types in Windows

Logon Type	Description	Purpose
Interactive	Logon locally	This logon type is intended for users who will be interactively using the computer, such as a user being logged on by a terminal server, remote shell, or similar process. This logon type has the additional expense of caching logon information for disconnected operations; therefore, it is inappropriate for some client/server applications, such as a mail server.
Network	Access this computer from the network	This logon type is intended for high performance servers. The LogonUser function does not cache credentials for this logon type
Service	Logon as a service	Indicates a service-type logon. The account provided must have the service privilege enabled.
Batch	Logon as a batch job	This logon type is intended for batch servers, where processes may be executing on behalf of a user without their direct intervention. This type is also for higher performance servers, such as mail or Web servers. The LogonUser function does not cache credentials for this logon type
Unlock	Unlock screen saver	This logon type is for WinLogon extension DLLs that log on users who will be interactively using the computer. This logon type can generate a unique audit record that shows when the workstation

Logon Type	Description	Purpose
		was unlocked.
New Credentials	Clone and create new security token	This logon type allows the caller to clone its current security token and specify new credentials for outbound connections. The new logon session has the same local identifier but uses different credentials for other network connections.
Network_ClearText	Anonymous authentication to IIS	This logon type preserves the name and password in the authentication package, which allows the server to make connections to other network servers while impersonating the client. A server can accept plaintext credentials from a client, call LogonUser, verify that the user can access the system across the network, and still communicate with other servers. When IIS is configured to not require a client to re-authenticate and assigns a specified account for users to be associated with the anonymous connection. In the evaluated configuration IIS will only accept request from authenticated clients

Each of the logon types has a corresponding user logon right that can be assigned to user and group accounts to control the logon methods available to users associated with those accounts.

6.1.4.2 Trusted Path and Re-authentication

For initial interactive logon, a user must invoke a trusted path in order to ensure the protection of identification and authentication information. The trusted path is invoked by using the **Ctrl+Alt+Del** key sequence, which is always captured by the TSF (i.e., it cannot be intercepted by an untrusted process), and the result will be a logon dialog that is under the control of the TSF. Once the logon dialog is displayed, the user can enter their identity (username and domain) and authentication (password or smartcard PIN). For remote logon, a user must initially do an interactive logon for which a trusted path is provided (as described above). Additionally, the TSF uses IPSec, among other techniques, to provide a trusted path between TSFs to ensure the protection of the I&A information transferred between TSFs.

A user can change their password either during the initial interactive log or while logged on. To change a user's password, the user must invoke the trusted path by using the **Ctrl+Alt+Del** key sequence. The logon dialog displayed allows the user to select an option to change their password. If selected, a change password dialog is displayed which requires the user to enter their current password and a new password. The TSF will change the password only if the TSF can successfully authenticate the user using the current password that is entered (see section Logon Process for a description of the authentication process) and if the new password conforms to the password policy defined by the administrator.

Another action that requires the user to invoke the trusted path by using the **Ctrl+Alt+Del** key sequence and re-authenticate themselves is session locking and unlocking (see the **Session Locking Function** section).

6.1.4.2.1 Logon Banner

An authorized administrator can configure the interactive logon screen to display a logon banner with a title and warning. This logon banner will be displayed immediately before the interactive logon dialog (see above) and the user must select "OK" to exit the banner and access the logon dialog.

Furthermore, when a user logs onto an interactive session, they are presented with the date and time of their last successful login along with the number of unsuccessful attempts that may have occurred since then. This information persists in a dialog that can be dismissed using the "OK" button.

6.1.4.3 User Attribute Database

6.1.4.3.1 User and Group Accounts Definitions

Windows machine maintains databases (collectively referred to as user attribute database) that fully define user and group accounts. These definitions include:

- Account name: used to represent the account in human-readable form
- SID: a User Identifier (UID) or group identifier used to represent the user or group account within the TOE
- Password (only for user accounts): used to authenticate a user account when it logs on (stored in hashed form and is encrypted when not in use using a Rivest's Cipher (RC)4 algorithm and a RC4 system generated key)
- Private/Public Keys: used to encrypt and decrypt user's FEK
- Groups: used to associate group memberships with the account
- Privileges: used to associate TSF privileges with the account
- Logon rights: used to control the logon methods available to the account (e.g. the "logon locally" right allows a user to interactively logon to a given system)
- Smart Card Policy: used to require a smart card to logon
- Miscellaneous control information: used to keep track of additional security relevant account attributes such as allowable periods of usage, whether the account has been locked, whether the password has expired, password history, and time since the password was last changed
- Other non-security relevant information: used to complete the definition with other useful information such a user's real name and the purpose of the account.

Note that security relevant roles are associated with users by virtue of group assignments (which in turn have privilege assignments) and are not otherwise specifically identified.

The actual composition of the user attribute database on each machine depends upon how that machine is configured (e.g., stand-alone, domain member, domain controller (DC)).

A standalone Windows machine, or a Windows machine with locally-defined accounts, uses the Security Accounts Manager (SAM) database as the user attribute database.

For managed network environments, Windows allows the establishment of domains that are managed by the Active Directory (AD). Active Directory domains enable a collection of Windows machines to share a common set of policies and accounts. These common accounts and policies are managed by the Active Directory Domain Services server role in Windows Server 2008.

6.1.4.4 Active Directory

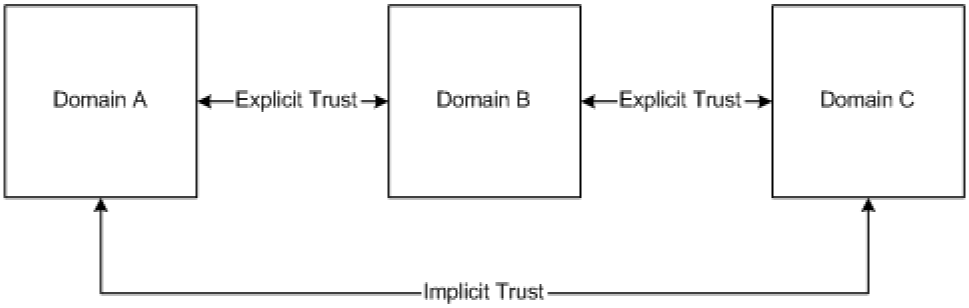
The Active Directory Domain Services role, also known as a Domain Controller (DC), instantiates AD services that define policies and accounts that are shared by Windows machines in the domain. In addition group policies (see **Security Management (FMT)**) can also be defined in the AD that apply to selected machines and accounts within the domain.

The topology of an Active Directory deployment can be described as a collection of AD domains, trees, forests, and trust relationships between domains and forests.

- **Tree**: A tree is a set of one or more Windows Server 2008 R2 domains sharing a common schema, configuration, and global catalog, joined together to form a contiguous namespace. All domains in a given tree trust each other through transitive hierarchical Kerberos trust relationships. A larger tree can be constructed by joining additional domains as children to form a larger contiguous namespace. Enterprise deployments of Active Directory can be a single-tree or a multi-tree. Naming within a given tree is always contiguous.
- **Forest**: A forest is a set of one or more trees that do not form a contiguous namespace. All trees in a forest share a common schema, configuration, and GC. All trees in a forest trust each other through transitive, hierarchical Kerberos trust relationships. Unlike trees, a forest does not need a distinct name. A forest exists as a set of cross-reference objects and Kerberos trust relationships known to the member trees. Trees in a forest form a hierarchy for the purposes of Kerberos trust; the tree name at the root of the trust tree can be used to refer to a given forest.

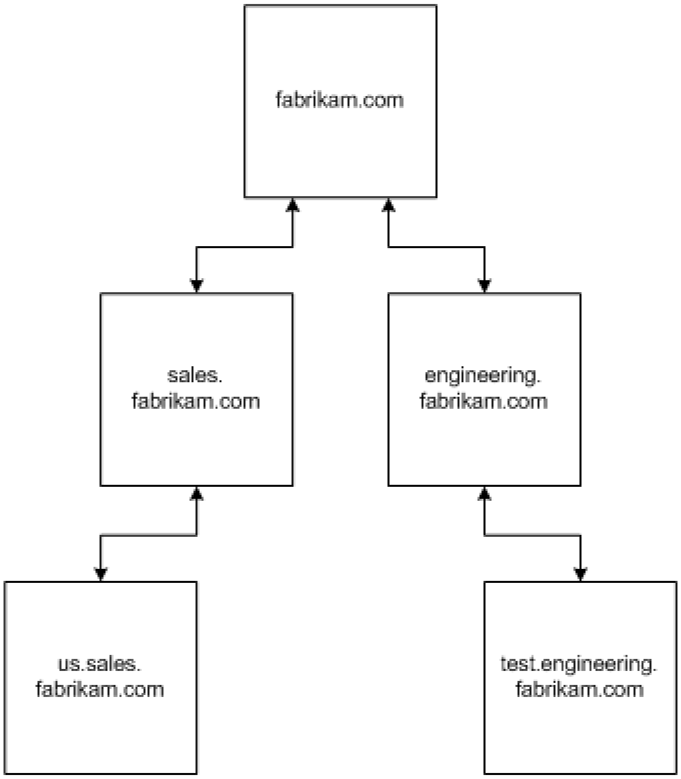
In a networked environment, an Active Directory user attribute database can be logically extended further through trust relationships. Each DC is configured to trust other domains that collectively comprise a common namespace. The result is that accounts from trusted domains can be used to access the trusting domain. These trust relationships between domains are based on the Kerberos security protocol. Kerberos trust is transitive and hierarchical—if domain A trusts domain B, and domain B trusts domain C, then domain A trusts domain C.

Figure 1 Trust Relationshipins in Active Directory



Since an AD directory information tree forms a contiguous namespace, another way to view the relationship between domains is based on the namespace. An object's distinguished name deinfes a path up the hierarchy of the domain tree namespace to create a grouping of objects into a logical hierarchy.

Figure 2 Domain Namespaces in Active Directory



To summarize, Windows uses the local SAM user account database to authenticate a locally-defined user account when either the machine is not joined to an AD domain or authenticating a local account (as designated by the <namespace>\username supplied during logon). Otherwise the Windows machine will use the Active Directory user account database for a logon using a AD account.

Refer to http://msdn.microsoft.com/en-us/library/cc223122(PROT.10).aspx for more information regarding the Windows implementation of Active Directory and associated domain and LDAP services.

6.1.4.5 Account Policies

Complementary to the user account database is the account policy that is defined on each TSF and in each domain. The account policy is controlled by an authorized administrator and allows the definition of a password policy and an account lockout policy with respect to interactive logons.

The password policy includes:

- The number of historical passwords to maintain in order to restrict changing passwords back to a previous value;
- The maximum password age before the user is forced to change their password;
- The minimum password age before the user is allowed to changed their password;
- The minimum password length when changing to a new password (0 or higher).
- Pre-defined password complexity requirements that can be enabled or disabled.

The account lockout policy includes:

- Duration (including an option for an indefinite lockout requiring an administrator to enable the account) of the account lockout once it occurs;
- Number of failed logon attempts before the account will be locked out;
- The amount of time after which the failed logon count will be reset.

These policies allow Windows to make appropriate decisions and change user attributes in the absence of an authorized administrator. For example, Windows will “expire” a password automatically when the maximum password age has been reached. Similarly, it will lock an account once a predefined number of failed logon attempts have occurred and will subsequently only unlock the account as the policy dictates. These policies also serve to restrict features available to authorized users (e.g., frequency of password change, size of password, reuse of passwords).

6.1.4.6 Logon Process

All logons are treated essentially in the same manner regardless of their source (e.g., interactive logon dialog, network interface, internally initiated service logon). They begin with an account name, domain name (which may be NULL; indicating the local system), and password that must be provided to the TSF.

The domain name indicates where the account is defined. If the local machine name (or NULL) is selected for the domain name, the local user account database is used. Otherwise the user account database associated with that machine’s Active Directory domain will be used. If the domain name provided does not match that of the DC, the DC will attempt to determine whether the target domain is a trusted domain. If it is, the trusted domain’s user account database will be used. Otherwise, the logon attempt will fail.

At this point, one of two types of authentication protocols will be used, either Windows Challenge / Response (NTLM) or Kerberos. Kerberos is the default logon method and will be used if a Kerberos KDC is available. Each DC includes a KDC in addition to its AD. NTLM will be used if no KDC is available or if the client requests NTLM authentication instead of Kerberos authentication. In the evaluated configuration a KDC is available to each DC.

There are two primary differences between NTLM and Kerberos logons. The first is that NTLM requires that the username and a hashed version of the password be sent, as part of a hashed response to a challenge, to the appropriate DC (or for a local account, the SAM database). The receiving TSF will compare the NTLM challenge response containing hashed password with the information stored in its database for the user identified by the username. If the hashed passwords match, authentication is successful. Kerberos, on the other hand, requires that a time-stamped logon request be encrypted with the hashed password. The encrypted request is sent to the appropriate DC, which in turn looks up the user's hashed password in its database. The hashed password is used to decrypt the logon request. If the decrypt operation succeeds and the logon request has an appropriate time stamp (i.e., within a time period set by an authorized administrator), authentication is successful. In either case, a successful authentication yields the user's SID and the SIDs of the user's groups as defined on the authenticating DC (or local TSF for a local account). Note that a failed authentication attempt increments the number of failed logon attempts for the user account and may result in the account being locked out (i.e., unable to logon).

The second primary difference between NTLM and Kerberos logon is in how subsequent requests for service (i.e., network logons) will occur. In the case of NTLM, the user must logon to every Windows machine in order to obtain a service (e.g., access to a file). These will be network logons and will essentially follow the same process as the initial interactive logon. A Kerberos logon yields a Ticket Granting Ticket (TGT) that is used to subsequently request Service Tickets from the KDC each time the user process wants to access a network service. The Service Ticket, containing some of the user's security attributes, will serve to authenticate the user rather than effectively requiring re-authentication using a hashed password. A third possibility exists in the case of Kerberos Protocol Transition.

The Windows Kerberos feature is an extended implementation of the Kerberos Network Authentication Service (V5) protocol (RFC 4120). The extension includes, for example, additional capabilities associated with group memberships, integrity levels, and delegation and encryption support. Refer to http://msdn.microsoft.com/en-us/library/cc233855(PROT.10).aspx for more information regarding the Windows implementation of Kerberos. Furthermore, the Windows Kerberos feature is designed to employ the applicable cryptographic protection mechanisms described earlier.

When a IIS service process, for example, is configured to be trusted for delegation and to accept other protocols than Kerberos authentication, it can authenticate users (e.g., using NTLM) and then will use the service process' Kerberos credentials to access content from other servers on behalf of the user. This is useful where the server may be behind a firewall where the user process cannot obtain a Kerberos TGT since it cannot access the KDC. As such, the user provides authentication information acceptable by the IIS server process and the IIS server process uses Kerberos to obtain its own TGT. Then, depending on the delegation level assigned to the server, the server can subsequently impersonate the user in order to perform operations on their behalf.

In any case once a successful authentication occurs, the Windows machine will query its AD (via its DC), if applicable, for group policies relevant to the user that is attempting to logon. Windows will use its user attributes database (including domain properties, such as from a group policy) to derive additional security attributes for the user (e.g., privileges and user rights). Windows will then ensure that any logon constraints defined in its user attributes database (including domain properties applicable to the user) to the user are enforced prior to completing a successful logon. If there are no constraints that would prevent a successful logon, a process (or thread, when the logon server is going to impersonate the user) is created and assigned a token that defines a security context based on the attributes collected during the logon process (user and group SIDs, privileges, logon rights, as well as a default DACL created by the logon process).

Note that if the User Account Control feature is enabled, the process of any user with authorized administrator access rights is initially assigned only those rights available to standard users. Subsequently, if that process attempts to perform an operation requiring the access rights of an authorized administrator, the user will be prompted to confirm whether the access right escalation should occur. If acknowledged, the full authorized administrator access rights are enabled in the process' token.

When a web site or another computer requests authentication through NTLM or Kerberos, an Update Default Credentials or Save Password check box appears in the Net Logon UI dialog box. If the user selects the check box, the Credential Manager keeps track of the user's name, password, and related information for the authentication service to use.

The next time that service is used, the Credential Manager automatically supplies the stored credential. If it is not accepted, the user is prompted for the correct access information. If access is granted, the Credential Manager overwrites the previous credential with the new one.

6.1.4.6.1 Smart Card Logon Processing

The TOE offers the ability to authenticate with a smart card in addition to authentication with a password. The smart card logon process begins when the user inserts a smart card into a smart card reader attached to the computer. When the TOE is configured for smart card logon, the insertion of the card signals the Secure Attention Sequence (SAS), just as the key combination **Ctrl+Alt+Del** signals the SAS on computers configured for password logon. In response, the TOE forces the display of a logon dialog box and the user is prompted to provide a PIN. Note that the PIN is required by the smart card, which is not part of the TOE. As such, it is assumed that users will physically protect their smart cards and the smart card requirement to provide a PIN for access serves only as an extra, unevaluated, mechanism offered by the TOE environment.

The user's logon information is sent to the LSA just as it does with a username/password logon. The LSA Kerberos authentication package uses the PIN for access, via the Smart Card Helper RPC Interfaces, to the smart card. The smart card contains the user's private key along with an X.509 v3 certificate that contains the public half of the key pair. The cryptographic operations that use these keys take place on the smart card.

After the initial private-key authentication, standard Kerberos protocols for obtaining session tickets are used to connect to network services. When the KDC is not available in the case of a smart card cached logon request, the verification information (e.g., supplemental credentials) is provided by the MSV1_0 authentication package.

The behavior of the TOE with respect to smart card removal is governed by a registry value which dictates which of the following actions will occur as a reaction to the removal of the smart card: no action, the workstation is locked, a logout is forced. If the workstation is locked, the user will be prompted to reinsert their smart card and enter the applicable PIN so that its contents can be verified before unlocking the workstation for use.

6.1.4.6.2 Network Logon Support

PK-certificate network logon is supported by the TLS/SSL Security Provider that implements the Microsoft Unified Security Protocol Provider security package. This package provides support for several network security protocols, and in particular SSL version 3.0, TLS versions 1.0, 1.1 and 1.2. In the TOE, security package APIs are not directly accessible, rather they are accessed via LSA Authentication APIs. The TLS/SSL Security Provider authenticates connections, and/or encrypts messages between clients and servers. When an application needs to use a network resource on an authenticated channel, the LSA accesses the TLS/SSL Security Service Provider (SSP) via the SSP interfaces. Windows implements TLS/SSL in accordance with RFCs 5246 and 2246 with extensions specified in RFCs 4366, 3546, and 4681 and additional supported cipher suites as specified in RFCs 3268, 4492, and 5289. For more information regarding the Windows implementation of TLS/SSL refer to http://msdn.microsoft.com/en-us/library/dd207968(PROT.10).aspx. Furthermore, the Windows TLS/SSL feature is designed to employ the applicable cryptographic protection mechanisms described earlier.

Digest network logon is supported by the Microsoft Digest Access Authentication Package. Digest performs user authentication for LSA Authentication in support of network logon attempts. Interactive logons cannot be performed using Digest Access. Digest implements a network security protocol, in this case digest challenge/response authentication that supports remote network logon user authentication and other network security services according to RFCs 2617 and 2831. For more information regarding the Windows implementation of Digest Access Authentication refer to http://msdn.microsoft.com/en-us/library/cc227906(PROT.10).aspx. Furthermore, the Windows Digest feature is designed to employ the applicable Cryptographic Protection mechanisms described earlier.

6.1.4.7 Impersonation

In some cases, specifically for server processes, it is necessary to impersonate another user in order to ensure that access control and accountability are performed in an appropriate context. To support this, the TSF includes the ability for a server to impersonate a client. As described above, each process has a token that primarily includes account SIDs, privileges, logon rights, and a default DACL. Normally, each thread within a process uses the process' token for its security context. However, a thread can be assigned an impersonation token that would be used instead of the processes token when making access checks and generating audit data. Hence, that thread is impersonating the client that provided the impersonation token. Impersonation stops when the impersonation token is removed from the thread or when the thread terminates.

When communicating with a server, the client can select an impersonation level that constrains whether and how a server may impersonate the client. The client can select one of four available impersonation levels: anonymous, identify, impersonate, and delegate:

- Anonymous allows the server to impersonate the client, but the impersonation token does not contain any client information.
- Identify allows the server to get the identity and privileges of the client, but can not impersonate the client.
- Impersonate enables the server to impersonate, i.e., perform access checks as the client's security context on the local system to access resources local to the server's TSF.
- Delegate enables server can impersonate the client's security context on local and remote systems.

6.1.4.8 Restricted Tokens

Whenever a process is created or a thread is assigned an impersonation token, Windows allows the caller to restrict the token that will be used in the new process or impersonation thread. Specifically, the caller can remove privileges from the token, assign a deny-only attribute to SIDs, and specify a list of restricting SIDs. That is:

- Removed privileges are simply not present in the resulting token.
- SIDs with the deny-only attribute are used only to identify access denied settings when checking for access, but ignore any access allowed settings.
- When a list of restricting SIDs is assigned to a token, access is checked twice once using the tokens enabled SIDs and again using the restricting SIDs. Access is granted only if both checks allow the desired access.

6.1.4.9 Strength of Authentication

As indicated above, Windows provides a set of functions that allow the account policy to be managed. These functions include the ability to define account policy parameters, including minimum password length. The minimum password length can be configured to require 16 characters. The administrator guide recommends a minimum password length adequate to ensure the metrics in the **Verification of Secrets (FIA_SOS.1)** requirement are satisfied. The administrator can also configure the number of passwords for Windows to remember so that a user cannot reuse a previous password until the password has changed the configure number of times.

During authentication, the Logon UI will not provide feedback that will reduce the probability of guessing a password beyond eliminating that once choice. However, if an account becomes locked, Windows will report that the account is disabled. Furthermore, the TSF forces a delay between attempts, such that there can be no more than ten attempts per minute.

For each subsequent failed logon following five consecutive failed logon occurrences in the last sixty seconds, the logon component sleeps for 30 seconds before showing a new logon dialog. It therefore supports the I&A function that no more than ten interactive logon attempts are possible in any sixty second period.

When Kerberos is used, the password requirements are the same as those described above. However, there are both Ticket Granting Tickets and Service Tickets that are used to store, protect, and represent user credentials and are effectively used in identifying and authenticating the user. Session keys are initially exchanged using a hash of the user’s password for a key.

SFR Mapping:

The **Identification and Authentication** function satisfies the following SFRs:

- FIA_AFL_EXT.1: The TSF locks the account after the administrator-defined threshold of unsuccessful logon attempts has occurred. The account will remain locked until an authorized administrator unlocks it. Note that the limit of 10 attempts per minute is enforced regardless of the threshold. While locked, responses to the user will not reflect whether the authentication attempt was successful.
- FIA_ATD.1: Each Windows machine has a user attribute database for local machine accounts. Each user attribute database describes accounts, including identity, group memberships, password (e.g., authentication data), privileges, logon rights, allowable time periods of usage, smart card policy, as well as other security-relevant control information. Security-relevant roles are associated with users via group memberships and privileges. Windows machines joined to an Active Directory domain utilize the Active Directory as the user attribute database for domain user and machine accounts.
- FIA_SOS.1: The administrator can configure a minimum 16 character password length and also the number of historical passwords to remember to prevent frequent password reuse.
- FIA_UAU.1: An authorized administrator can configure the TSF to allow no TSF-mediated functions prior to authentication, with the exception of access to the web server.
- FIA_UAU.6: The TSF will only allow a password to be changed if the TSF can successfully authenticate the user using the current password which must be entered with the new password.
- FIA_UAU.7: During an interactive logon, the TSF echoes the users password with “*” characters to prevent disclosure of the user’s password.
- FIA_UID.1: An authorized administrator can configure the TSF to allow no TSF-mediated functions prior to identification, with the exception of access to the web server.
- FIA_USB.1: Each process and thread has an associated token that identifies the responsible user (used for audit and access), associated groups (used for access), privileges, Mandatory Integrity Control integrity labels and policies, and logon rights held by that process or thread on behalf of the user. Additionally, a public/private key pair is associated with a user’s account when a user encrypts a file, and an authorized administrator can assign a public/private key pair to a user account. Normally the security attributes assigned to a process and its threads remain

unchanged, but when User Account Control is enabled, processes belonging to an authorized administrators are initially assigned an access token limited to access rights available to other standard users and must interactively acknowledge the escalation before the process can use the full authorized administrator access rights. Note that any changes to user security attributes are applied when the user next logs in and a new subject is created to act on the user's behalf.

- FTA_MCS.1: By enforcing the allow/deny local logon right, user accounts can be restricted to specific workstations in the domain thereby enforcing the maximum number of interactive concurrent sessions per user based upon those machines the authorized administrator has defined an account upon for any given user.
- FTA_TAB.1, FMT_MTD.1a (partially implemented – see section **6.1.5** for the rest of the implementation): An authorized administrator can define and modify a banner that will be displayed prior to allowing a user to logon.
- FTA_TAH.1: When a user logs onto an interactive session, they are presented with a dialog with the date and time of their last successful login and the number of unsuccessful attempts that may have occurred since then. This dialog can be dismissed by the user.
- FTA_TSE.1: The TSF will not allow a user to logon if the user's password has expired. The TSF will restrict the location a user can logon from based upon the logon rights associated with a user's account (logon locally, logon as a batch job, access this computer from the network, and logon as a service). Additionally, the TSF restricts a user from logon based upon time or day in that a user will not be able to logon if attempts are made after an account has been locked out but within the account lockout duration defined by the authorized administrator.
- FTP_TRP.1: The TSF provides an unspoofable key sequence, **Ctrl+Alt+Del**, that can be used to assure that the user is communicating directly with the TSF for purposes of initial interactive logon (with password) and session unlocking. When the TOE is configured for smart card logon, the insertion of the card signals the SAS, just as the key combination **Ctrl+Alt+Del** signals the SAS on computers configured for password logon.

6.1.5 Security Management Function

The TOE supports the definition of roles as well as providing a number of functions to manage the various security policies and features provided by the TOE.

6.1.5.1 Roles

The notion of a role within the TOE is generally realized by assigning group accounts and privileges to a given user account. Whenever that user account is used to logon, the user will be assuming the role that corresponds with the combination of groups and privileges that it holds. While additional roles could be defined, this ST defines the authorized administrator role as being special.

The Administrator role is defined as any user account that is assigned one of the security-relevant privileges (e.g., Take Owner privilege) or is made a member of one or more of the several pre-defined administrative groups (e.g., Administrators, Cryptographic Operators, and Backup Operators local groups). The Administrator Guide fully identifies all security-related privileges and administrative groups, and provides advice on how and when to assign them to user accounts. A user assumes an administrator role by logging on using a user account assigned one of these privileges or group membership.

Any user that can successfully logon is considered to be in an authorized user, though this is not specifically identified as a security management role per se. Of the functions all users can perform, creating objects, modifying DAC permissions of their objects, and managing their own passwords are particularly notable.

6.1.5.2 Security Management Functions

The TOE supports a number of policies and features that require appropriate management. With few exceptions, the security management functions are restricted to an authorized administrator. This constraint is generally accomplished by privilege or access control (e.g., SD), and occasionally by a specific SID requirement (e.g., "Administrators"). The TOE supports security management functions for the following security policies and features:

- **Audit Policy**: The audit policy management functions allow an authorized administrator the ability to enable and disable auditing, to configure which categories of events will be audited for success and/or failure, and to manage (e.g., clear) and access the security event log. An authorized administrator can also define specifically which user and access mode combinations will be audited for specific objects in the TOE.
- **Account Policy**: The account policy management functions allow only an authorized administrator to define constraints for passwords (password complexity requirements), account lockout (due to failed logon attempts) parameters, and Kerberos key usage parameters. The constraints for passwords restrict changes by including minimum password length, password history, and the minimum and maximum allowable password age. If the maximum password age is exceeded, the corresponding user cannot logon until the password is changed. The account lockout parameters include the number of failed logon attempts (in a selected interval) before locking the account and duration of the lockout. The Kerberos key usage parameters primarily specify how long various keys remain valid. While an authorized administrator can change passwords and a user can change their own password, the TSF does not allow any user (including the authorized administrator) to read passwords. Additionally, the authorized administrator can define the advisory warning message displayed before access to the TOE is granted.
- **Account Database Policy**: The account database management functions allow an authorized administrator to define, assign, and remove security attributes to and from both user and group accounts, both locally and for a domain, if applicable. The set of attributes includes account names, SIDs, passwords, group memberships, and other security-relevant and non-security relevant information. Of the set of user information, only the password can be modified by a user that is not an authorized administrator. Specifically, an authorized administrator assigns an initial password when an account is created and may also change the password like any other account attribute. However, a user may change their password. This is enforced by requiring the user to enter their old password in order to change the password to a new value.

- **User Rights Policy**: The user rights management functions allow an authorized administrator to assign or remove user and group accounts to and from specific logon rights and privileges.
- **Domain Policy**: The domain management functions allow an authorized administrator to add and remove machines to and from a domain as well as to establish trust relationships among domains. Changes to domains and domain relationships effectively change the definition and scope of other security databases and policies (e.g., the account database). For example, accounts in a domain are generally recognized by all members of the domain. Similarly, accounts in a trusted domain are recognized in the trusting domain.
- **Group Policy**: The group policy management functions allow an authorized administrator to define accounts, user right assignments, and TOE machine/computer security settings, etc. for a group of TSFs or accounts within a domain. The group policies effectively modify the policies (e.g., machine security settings, and user rights policy) defined for the corresponding TSFs or users.
- **IPSec Policy**: The IPSec management functions allow an authorized administrator to define whether and how (e.g., protocols and ports to be protected, outbound and/or inbound traffic, with what cryptographic algorithms) IPSec will be used to protect traffic among distributed TSFs.
- **EFS Policy**: The EFS management functions allow an authorized administrator to enable or disable EFS on an NTFS volume and generally control the recovery for EFS data.
- **Disk Quota Policy**: The disk quota management functions allow an authorized administrator to manage disk quotas for NTFS volumes. More specifically, the functions allow an authorized administrator to enable or disable disk quotas, define default disk quotas, and define actions to take when disk quotas are exceeded.
- **DAC Policy**: The DAC functions allow authorized users to modify access control attributes associated with a named object.
- **FEK Policy** - The first time a user encrypts a file, the TSF assigns the user account a public/private key pair which is used to protect the randomly generated FEK associated with the file. Only the owner of the private key used to protect the FEK associated with the file, or an administrator or subject with a specific privilege, can delete the FEK.
- **Other**: The TSF also allows the administrator the ability to modify the time and modify object integrity labels.

6.1.5.3 Valid Attributes

The TSF ensures that only valid values are accepted as security attributes for the password. Valid values are values that are meet the password complexity restrictions as defined by the administrator. For example, the minimum password length should be set to greater than or equal to eight characters by the administrator. Subsequently, attempts to create passwords shorter than eight characters will not be accepted by the TSF.

Beyond this, the TSF generally checks parameters provided for security management and other functions in order to ensure that only valid values are accepted in order to avoid the potential to get into unknown or bad states of operation.

SFR Mapping;

The **Security Management** function satisfies the following SFRs:

- FMT_MOF.1a: Only an authorized administrator can enable and disable the audit mechanism, select which audit events will be recorded in accordance with FAU_SEL.1.
- FMT_MOF.1b: Only an authorized administrator can configure the settings that serve to constrain acceptability of authentication data (length, history, complexity, etc.).
- FMT_MSA.2: The TSF ensures that values for password security attributes meet the password complexity and other restrictions, as defined by the administrator. Furthermore, each security management function is generally designed to ensure that values offered by administrators are valid before being accepted.
- FMT_MTD.1a: As a rules security management functions are limited to authorized administrators as indicated above. Manipulation of a user's own authentication data is a notable exception.
- FMT_MTD.1b: Only an authorized administrator can view or clear the security event log. Furthermore, only authorized administrators can manipulate the security event log to cause applicable files to be archived or deleted.
- FMT_MTD.1c, FMT_MTD.1e: Only an authorized administrator can initially assign a password to a user account. Subsequently, both an authorized administrator and the user corresponding to the password can change a password.
- FMT_MTD.1d: Only an authorized administrator can define user accounts and group accounts, define user/group associations (e.g., group memberships), assign privileges and user rights to accounts, as well as define other security-relevant and non-security relevant user attributes, with the exception of passwords (which are addressed above) and private/public key pairs.
- FMT_MTD.1f: Using access controls and one-way hashing of authentication data, no users are able to read authentication data.
- FMT_MTD.1g: Only an authorized administrator can manage cryptographic parameters that define how the TSF will use available cryptographic functions.
- FMT_REV.1a: Only an authorized administrator can remove security attributes from users and group accounts. By default such changes take effect the next time the user attempts to log in.
- FMT_SAE.1: Only an authorized administrator can set account policy parameters, including the maximum allowable password age before the account will be unable to logon. Once a password has expired, the TSF can be configured to require that the password be changed prior to successfully logging in.
- FMT_SMF.1: The TSF provides the administrator with, among other things, the capability to modify the time and object integrity labels and define the following policies:
    - Account Database Policy
    - Account Policy
    - Audit Policy
    - DAC Policy
    - Disk Quota Policy

  - Domain Policy
  - IPSec Policy
  - EFS Policy
  - File Encryption Key Policy
  - Group Policy
  - User Rights Policy

  Specifically, the TSF provides the administrator with the capability to perform the following:

  - Account Database Policy
    - initialize and modify user security attributes
  - Account Policy
    - modify the behavior of the locked user session function
    - modify the duration the user account is disabled after the unsuccessful authentication attempts threshold is exceeded
    - modify the minimum allowable password length
    - modify the advisory warning message displayed before establishment of a user session
    - modify the password complexity restriction
    - modify the unsuccessful authentication attempts threshold
  - Audit Policy
    - enable, disable, modify the behavior of the audit function and clear the audit trail
    - modify the set of events to be audited
    - read the audited events
    - modify the audit log size
  - DAC Policy
    - modify access control attributes associated with a named object
  - Disk Quota Policy
    - modify the quota settings on NTFS volumes
  - File Encryption Key Policy
    - delete encryption policy attributes associated with a file
    -
  - IPSec Policy
    - determine and modify the behavior of the function that protects data during transmission between parts of the TOE
    -

- FMT_SMR.1: The TOE supports the definition of an authorized administrator through the association of specific privileges and group memberships with user accounts. As described in the User Data Protection section, users are generally allowed to control the security attributes of objects depending upon the access that they have to those objects. Users can also modify their own authentication data (e.g., passwords) by providing their old password for authorization. Additionally, upon the creation of an object, the user creating the object (object creator) can define initial values for its security attributes that override the default values (e.g. DACL).

6.1.6 TSF Protection Function

The TSF provides a security domain for its own protection and provides process isolation. The security domains used within and by the TSF consists of the following:

- Hardware
- Kernel-mode software
- Trusted user-mode processes
- User-mode Administrative tools process

The TSF hardware is managed by the TSF kernel-mode software and is not modifiable by untrusted subjects. The TSF kernel-mode software is protected from modification by hardware execution state and memory protection. The TSF hardware provides a software interrupt instruction that causes a state change from user mode to kernel mode. The TSF kernel-mode software is responsible for processing all interrupts, and determines whether or not a valid kernel-mode call is being made. In addition, the TSF memory protection features ensure that attempts to access kernel-mode memory from user mode results in a hardware exception, ensuring that kernel-mode memory cannot be directly accessed by software not executing in the kernel mode.

The TSF provides process isolation for all user-mode processes through private virtual address spaces (private per process page tables), execution context (registers, program counters), and security context (handle table and token). The data structures defining process address space, execution context and security context are all stored in protected kernel-mode memory. All security relevant privileges are considered to enforce TSF Protection.

User-mode administrator tools execute with the security context of the process running on behalf of the authorized administrator. Administrator processes are protected like other user-mode processes, by process isolation.

Like TSF processes, user processes also are provided a private address space and process context, and therefore are protected from each other. Additionally, on 64-bit based hardware platforms, the TSF has the added ability to protect memory pages using Hardware Data Execution Prevention (DEP). Hardware-enforced DEP marks all memory locations in a process as non-executable unless the location explicitly contains executable code. Hardware-enforced DEP relies on processor hardware to mark memory with an attribute that indicates that code should not be executed from that memory location. DEP functions on a per-virtual memory page basis, usually by changing a bit in the page table entry (PTE) to mark the memory page. Processors that support hardware-enforced DEP are capable of raising an exception when code is executed from a page marked with the appropriate attribute set.

The TSF implements cryptographic mechanisms within a distinct user-mode process, where its services can be accessed by both kernel- and user-mode components, in order to isolate those functions from the rest of the TSF to limit exposure to possible errors while protecting those functions from potential tampering attempts.

Furthermore, the TSF includes a Code Integrity Verification feature, also known as Kernel-mode code signing (KMCS), whereby device drivers will be loaded only if they are digitally signed by either Microsoft or from a trusted root certificate authority recognized by Microsoft. KMCS uses public-key cryptography technology to verify the digital signature of each driver as it is loaded. When a driver tries to load, the TSF decrypts the hash included with the driver using the public key stored in the certificate. It then verifies that the hash matches the one that it computes based on the driver code using the FIPS -certified cryptographic libraries in the TSF. The authenticity of the certificate is also checked in the same way, but using the certificate authority's public key, which must be configured in and trusted by the TOE.

6.1.6.1 BitLocker Drive Encryption

In addition to protecting the TSF during runtime, BitLocker Drive Encryption (BDE) is a data protection feature available in Windows. It is responsible for helping prevent unauthorized access to data on lost or stolen systems (i.e., where physical access to the disk drive is possible) and therefore is intended primarily to defend against offline and online attacks when the user no longer has physical possession of the machine. BitLocker accomplishes this by combining two major data-protection procedures:

- Encrypting the entire Windows operating system volume on the hard disk.
- Verifying the integrity of early boot components and boot configuration data.

BitLocker protects hard drive data by providing Secure Startup (integrity checking of early boot components) and Full Volume Encryption (FVE). FVE protects data by encrypting entire disk volumes; in the case of the Windows operating system volume, this includes the swap and hibernation files. Secure Startup provides integrity checking of the early boot components, ensuring that FVE decryption is performed only if those components are found to be unchanged and the encrypted drive is located in the original computer.

BitLocker should be configured to use a Trusted Platform Module (TPM 1.2) to protect user data (e.g., by storing the applicable encryption keys) and to ensure that a PC running Windows has not been tampered with while the system was offline. BitLocker can be used without a TPM, however in such cases the secure startup protection cannot be utilized. Offline protection is provided by encrypting the entire Windows operating system volume, including both user and system files, the hibernation file, the page file, and temporary files. BitLocker implementations using TPM 1.2 help ensure the integrity of the startup process by:

- Providing a method to check that early boot file integrity has been maintained, and help ensure that there has been no adversarial modification of those files, such as with boot sector viruses or rootkits.
- Enhancing protection to mitigate offline software-based attacks. Any alternative software that might start the system does not have access to the decryption keys for the Windows operating system volume.

- Locking the system (i.e., by not releasing the necessary encryption keys) when tampered with, and if any monitored files have been tampered, the system does not start.

BitLocker optionally leverages an enterprise's existing Active Directory Domain Services infrastructure to remotely escrow FVE recovery keys and TPM ownership information.

On computers with TPM 1.2, BitLocker offers the option for multi-factor authentication, locking the normal boot process until the user supplies a PIN and/or inserts a USB device that contains keying material. It uses the TPM to perform system integrity checks on critical early boot components. The TPM collects and stores measurements (hashes) from multiple early Windows boot components and boot configuration data to create a system identifier for that computer. This is done so that if any early boot components are changed or tampered with the TPM will prevent BitLocker from unlocking the encrypted volume and will force the computer to enter recovery mode and will not unlock the protected volume until the TPM verifies system integrity; the computer will not boot or resume from hibernation until the correct PIN and/or USB device is presented.

BitLocker implementations on computers without TPM 1.2 can still be used to encrypt the Windows operating system volume. However, this implementation will require a USB startup key to start the computer or resume hibernation, and does not provide the pre-startup system integrity verification offered by BitLocker working with a TPM.

By default BitLocker will encrypt the hard drive using AES128-CBC with Elephant Diffuser; in addition, it can be configured to use (regular) AES-CBC using both 128 and 256 bit disk encryption keys. In the AES-CBC scenario, BitLocker derives the IV for a block from the AES key and block offset.

6.1.6.1.1 BitLocker To Go

The BitLocker function can also be employed to protect removable USB storage devices using a function called BitLocker To Go (BTG). While BitLocker To Go can be used for any removable USB storage device, an authorized administrator can configure (using the Group Policy mechanism) the TSF to require BitLocker To Go to be used in order to write data to such a device. If BitLocker To Go is required, devices that are not protected with this feature can only be accessed read-only.

When being used, the removable volume is assigned a Volume Master Key (VMK) generated using available cryptographic support functions. The VMK is encrypted using AES-CBC and depending on the machine configuration, either a user-provided password or a smartcard credential to generate a key that is stored on the removable USB storage device. A File Volume Encryption Key (FVEK) is also generated and encrypted using 256-bit AES using the VMK, and also stored on the removable USB storage device. Subsequently, all data encryption and decryption functions, in order to access the user-data content of the removable USB storage device, are performed using 128-bit AES-CBC using the FVEK.

In order to access the content of the removable USB storage device, a compatible Windows system can be used for full access provided the appropriate user credentials are available. Alternately, an available BitLocker Reader application (placed on a removable USB device when first configured to use this feature) can be used in conjunction with the proper user credentials in order to decrypt and copy content from the removable USB device, but this application does not support writing content to the device.

6.1.6.2 Internal TOE Protection

The TOE protects against unauthorized disclosure and modification of data when it is transferred between physically separated parts of the TOE using a suite of Internet standard protocols including IPSec and ISAKMP. IPSec can be used to secure traffic using IP addresses or port number between two computers. IPSec does not apply to broadcast or multicast traffic. IPSec services are configurable on the system to allow for a variety of security services including data origin authentication, message integrity, and data confidentiality. The TOE implements IPSec with a set of kernel subsystems and user-mode trusted servers. IPSec allows for the application of a set of security services to be applied to IP data based on predefined IPSec policies. The TOE stores IPSec and related key exchange protocol policies in the DS. At system initialization, these policies are retrieved and stored in the system registry and passed to the IPSec network driver. The TSF monitors for policy updates and processes these as well, by updating the system registry and updating the policy entries in the network driver as appropriate (modify, add, and delete). IPSec policies specify the functions that IPSec must perform for a given outbound or inbound packet. IPSec policies identify the local host algorithms and associated attributes, mode of communication (transport is the only mode included in the evaluation configuration), and a list of filters to be applied to IP packet traffic. Filters are used to associate inbound and outbound packets with a specific IPSec policy. They specify the source and destination IP addresses, ports, and protocol. IPSec uses the elliptic curve Diffie-Hellman (ECDH) to provide data confidentiality and integrity for IP packets.

Keys are exchanged between computers within the TOE before secured data can be exchanged by the establishment of a security agreement between the two computers. In this security agreement, called a Security Association (SA), both agree on how to exchange and protect information. To build this agreement between the two computers, the Internet Engineering Task Force (IETF) has established a standard method of security association and key exchange resolution named IKE which is applied in the TOE. A SA is the combination of a negotiated key, security protocol, and Security Parameters Index (SPI), which collectively define the security used to protect the communication from sender to receiver. The SPI is a unique identifying value in the SA that is used to distinguish among multiple SAs that exist at the receiving computer.

In order to ensure successful and secure communication, IKE performs a two-phase operation. Confidentiality and authentication are ensured during each phase by the use of encryption (i.e., AES per FCS_COP.1a) and authentication algorithms that are agreed upon by the two computers during security negotiations.

The IPSec management functions allow an authorized administrator to define the IPSec Policy including whether and how (i.e., protocols and ports to be protected, outbound and/or inbound traffic, with what cryptographic algorithms) IPSec will be used to protect traffic among distributed TSFs.

The evaluated configurations support the use of Kerberos and the use of public key certificate for machine authentication in the IKE processing. IKE processing includes the validation of the peer's certificate (including path validation) and signature payload verification.

The IPSec policy MMC snap-in allows an administrator to select the authentication method based on public key certificate. To use a public key certificate for authentication services the CA associated with the public key certificate and the associated root CA can be chosen. IKE processing maps a computer certificate to a computer account in an AD domain or forest, and then retrieves an access token, which includes the list of user rights assigned to the computer. An administrator can restrict access by configuring Group Policy security settings and assigning either the *Access this computer from the network* user right or the *Deny access to this computer from the network* user right to individual or multiple computers as needed.

The IKE processing also processes ISAKMP payload messages to allow IKE processing to obtain each other's public key value. IPSec policies and filters may be configured to reject the packet or audit the event if the results of a service applied to a packet challenges the integrity of the packet (modification, insertion of data, replay of data).

Windows implements the IKE protocol versions 1 and 2 as specified RFCs 2407, 2408, 2409, 3947, and 4306 with some extensions that provide additional capabilities, for example, related to Network Address Translation (NAT), fragmentation of large messages, and IPSec interoperation. Refer to http://msdn.microsoft.com/en-us/library/cc233219(PROT.10).aspx for specific information about the Windows implementation of IKE. Furthermore, the Windows IKE feature is designed to employ the applicable Cryptographic Protection mechanisms described earlier.

6.1.6.3 TSF Failure Recovery

When a failure occurs within the TSF, the TSF will immediately halt and produce a memory dump to a location on the system volume that is readable only be an authorized administrator. The machine will remain in a halted state until user intervention occurs. A user can then reset the system in order to reboot the operating system. During the subsequent boot, the user will be presented the option of booting into a limited mode (e.g., where only some device drivers and services are loaded or started) in order to attempt any necessary recovery functions (after logging in).

6.1.6.4 TSF Data Replication Consistency

In general, directory data resides in more than one place on the network. Through replication, the directory service maintains replicas of directory data on multiple DCs, ensuring directory availability and performance for all users. AD uses a multi-master replication model, allowing authorized users to make directory changes at any DC, not just at a designated primary DC.

The AD service allows for specific data to be replicated within the TOE. The AD namespace includes a directory information tree structure to facilitate the management of large size installations. Additionally, the AD includes the Global Catalog (GC), which is a partial index of select objects in the domain tree, combined with a search engine. The GC server returns the location of an object based on an object attribute provided by the user.

- **Tree**: A tree is a set of one or more Windows Server 2008 R2 domains sharing a common schema, configuration, and GC, joined together to form a contiguous namespace. All domains in a given tree trust each other through transitive hierarchical Kerberos trust relationships. A larger tree can be constructed by joining additional domains as children to form a larger contiguous namespace. Enterprises can be a single-tree or a multi-tree. Naming within a given tree is always contiguous.
- **Forest**: A forest is a set of one or more trees that do not form a contiguous namespace. All trees in a forest share a common schema, configuration, and GC. All trees in a forest trust each other through transitive, hierarchical Kerberos trust relationships. Unlike trees, a forest does not need a distinct name. A forest exists as a set of cross-reference objects and Kerberos trust relationships known to the member trees. Trees in a forest form a hierarchy for the purposes of Kerberos trust; the tree name at the root of the trust tree can be used to refer to a given forest.
- **GC server**: A GC server is a DC that stores specific information about all objects in a forest. The GC stores a replica of every directory partition in the forest. It stores full replicas of the schema and configuration directory partitions, a full replica of the domain directory partition for which the DC is authoritative, and partial replicas of all other domain directory partitions in the forest. When an “attributeSchema” object has the “isMemberOfPartialAttributeSet” attribute set to “TRUE,” the attribute is replicated from the domain directory partition to the corresponding directory partition replicas on all authoritative DCs and also to all GC Servers.

Any DC within a forest potentially could be a replication partner of another. Replication partners are determined by a replication topology. A replication topology is a set of AD connections by which DCs in a forest communicate over the network to synchronize the directory partition replicas that they have in common.

The replication topology determines the replication partnerships between source and destination DCs. As a replication source, the DC must determine the replication partners it must notify when changes occur. As a replication destination, the domain controller participates in replication either by responding to notification of changes from a source, or by requesting changes to initiate replication when it starts up or in response to a schedule.

The Knowledge Consistency Checker (KCC) is an element of AD that creates the replication topology. It creates connection objects on destination DCs that represent the inbound connection from the replication source DC. 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 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 DC receives a request for changes from a new destination, the source creates an entry for the destination in the “repsTo” attribute on the respective directory partition object.

Whenever the source has changes, it sends a notification to all replication partners that are identified in the “repsTo” value for the respective directory partition. Like the “repsFrom” data, this information is stored locally on the 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.

There are two types of TSF data replicated consistently throughout the TOE. They consist of Group Policy Objects (GPOs) and Directory Store (DS) data. GPOs are used to define configurations for groups of users and computers. GPOs store Group Policy information in two locations: a Group Policy Container (GPC) and a Group Policy Template (GPT). A GPC is a DS container that stores GPO properties that have settings in the GPO. As a DS Container the Group Policy Container is replicated throughout the domain with the rest of the DS data.

A GPT is a folder structure that stores Administrative Template-based policies, security settings, and applications available for software installation, and script files. When adding, removing, or modifying the contents of the SYSVOL folder on a DC, those changes are replicated to the SYSVOL folders on all other DCs in the domain. SYSVOL content uses the same replication schedule as the DS for inter-site replication.

Along with the GPO, all DCs contain three types of DS data: domain, schema, and configuration. In the case of the GC server a forth category consisting of a partial replica of domain data for all domains is added. Each type of data is separated into distinct directory partitions that form the basic units of replication for the DS. These partitions are as follows:

- **Domain partition**: all objects in the directory for a given domain; the data is replicated to every domain controller in that domain, but not beyond its domain.
- **Schema partition**: all object types (with attributes) that can be created in AD; the data is common to all domains in the domain tree or enterprise, and replicated to all DCs in the enterprise.
- **Configuration partition**: replication topology and related metadata; the data is common to all domains in the domain tree or enterprise, replicated to all DCs in the enterprise.

GC server also contains:

- **Domain data (partial replica) for all forest domains**: a read-only partial replica of the domain directory partition for all other domains in the enterprise and contains a subset of the properties for all objects in all domains in the enterprise.

The DS is a multi-master enabled database. This means that changes occur at any DC in the enterprise. This introduces the possibility of conflicts that can potentially lead to problems once the data is replicated to the rest of the enterprise. The DS addresses these potential conflicts in two ways.

One way, is by having a conflict resolution algorithm handle discrepancies in values by resolving to the DC to which changes were written last (that is, "the last writer wins"), while discarding the changes in all other DC’s.

For specific instances when conflicts are too difficult to resolve using the "last writer wins" approach, the DS updates certain objects in a single-master fashion. In a single-master model, only one DC in the entire directory is allowed to process updates. For management flexibility, this model is extended to include multiple roles, and the ability to transfer roles to any DC in the enterprise. This extended model is referred to as Flexible Single Master Operation (FSMO). In Windows 7 and Windows Server 2008 R2 there are four FSMO roles:

- **Schema master**: the single DC responsible for performing updates to the directory schema.
- **Domain naming master**: the DC responsible for making changes to the forest-wide domain name space of the directory. It can also add or remove cross-references to domains in external directories.
- **Relative Identifier (RID) master**: the single DC responsible for processing RID Pool requests for certain unique security identifiers from all DCs within a given domain. Users, computers, and groups that are stored in AD are assigned SIDs, which are unique alphanumeric numeric strings that map to a single object in the domain. SIDS consist of a domain-wide SID concatenated with a monotonically-increasing RID that is allocated by each DC in the domain. Each DC is assigned a pool of RIDs.
- **Infrastructure daemon**: the DC responsible for updating an object's SID and distinguished name in a cross-domain object reference.

The first two FSMO roles must be unique within a forest. The last two must be unique within each domain within a forest.

DS replication is not based on time, but on Update Sequence Numbers (USNs). Each DC holds a table containing entries for its own USN and the USNs of its replication partners. During replication, the DC compares the last known USN of its replication partner (saved in the table), with the current USN that the replication partner provides. If there have been recent changes (that is, if the replication partner provides a higher USN), the data store requests all changes from the replication partner (this is known as pull replication). After receiving the data, the directory store sets the USN to the same value as that of the replication partner.

If properties on the same object are changed on different DCs, the DCs reconcile the data by property version number, then by time stamp if the version numbers are the same, then by comparing the buffer size of a binary memory copy operation performed on each property. If the two buffers are equal, the attributes are the same, one is discarded.

Note that all reconciliation operations are logged, and authorized administrators have the option of recovering and using the rejected values.

6.1.6.5 Time Service

Each hardware platform supported by the TOE includes a real-time clock. The real-time clock is a device that can only be accessed using functions provided by the TSF. Specifically, the TSF provides functions that allow users, including the TSF itself, to query and set the clock, as well as functions to synchronize clocks within a domain. The ability to query the clock is unrestricted, while the ability to set the clock requires a privilege dedicated to that purpose. This privilege is only granted to authorized administrators to protect the integrity of the time service.

Each clock may be subject to some amount of error (e.g., “drift”), and management of that error is a topic in the administrator guidance. Additionally, since it may be important to have temporal correspondence across systems within a single domain, the TSF includes a domain clock synchronization function. One of the DCs is designated to provide the reference time. All clients (including other DCs) within the domain periodically contact the reference DC to adjust their local clock. The time between synchronization actions depends on the deviation between the local and reference clock (i.e., the more deviation, the sooner the next synchronization will be scheduled).

6.1.6.6 Network Access Protection

This feature allows access to network resources to be controlled based on a computer’s identity and compliance with configurable governance policies. The NAP mechanism is capable of automatically bringing a client workstation or server into compliance with defined governance policies so that access is subsequently allowed.

The NAP feature involves a NAP agent running on NAP clients and a NAP health policy server (NAP server) running on a Windows 2008 R2 server with the Network Policy Server (NPS) role.

The NAP agents collect relevant health information, including TSF data, installed application data, and other information (e.g., such as virus definition details) for their host NAP client and provides the health information to the NAP server when network access is required.

The NAP server uses NPS policies and settings, configured by an authorized administrator, to evaluate the health of NAP clients in order to determine whether to grant network access (full or restricted). When a NAP client is not conformant with configured settings and policies, only restricted network access would be allowed (i.e., that already available without any actions taken by the NAP server), but the NAP server and NAP agent can cooperate to remedy some identified problems in order to bring a NAP client into compliance so that its network access can be elevated.

For compliant clients, access to a network subsequent to NAP server approval can be granted by providing applicable network access credentials for the configured enforcement mechanism(s): IPsec, 802.1X, VPN, DHCP, and NAP-NAC (note that 802.1X and NPA_NAC are excluded since the TOE doesn’t include third party networking products).

6.1.6.7 TSF Code Integrity

The TSF Boot Manager is an Authenticode signed image file, based on the Portable Executable (PE) image file format. A SHA hash based signature and a public key certificate chain are embedded in the boot manager Authenticode signed image file under the “Certificate” IMAGE_DATA_DIRECTORY of the IMAGE_OPTIONAL_HEADER of the file. This public key certificate chain ends in a root public key. The boot manager uses the embedded SHA hash based signature and public key certificate chain to validate its own integrity. A SHA hash of the boot manager image file is calculated for the whole file, with the exception of the following three elements which are excluded from the hash calculation: the CheckSum field in the IMAGE_OPTIONAL_HEADER, the IMAGE_DIRECTORY_ENTRY_SECURITY IMAGE_DATA_DIRECTORY, and the public key certificate table, which always resides at the end of the image file.

If the boot manager is validated, then the root public key of the embedded public key certificate chain must match one of the Microsoft root public keys which indicate that Microsoft is the publisher of the boot manager. These root public keys are necessarily hardcoded in the boot manager. If the boot manager cannot validate its own integrity, then the boot manager does not continue to load other modules and displays an error message.

After the boot manager determines its integrity, it attempts to load one application from the following list of boot applications:

- Winload.exe or Winload.efi, the boot application used to load the Windows 7/WS08R2 kernel
- ntoskrnl.exe, the Windows kernel
- winresume.exe or winresume.efi, the boot application used for resuming from the hibernation file “hiberfil.sys”
- memtest.exe, a memory testing application.

These boot applications are also Authenticode signed image files. For each of the Windows 7/WS08R2 boot applications, the boot manager uses the embedded trusted SHA hash based signature and public key certificate chain within the boot application’s IMAGE_OPTIONAL_HEADER to validate the integrity of the boot application before attempting to load it. Except for the following three elements which are excluded from the hash calculation, a SHA hash of a boot application image file is calculated for the whole file: the CheckSum field in the IMAGE_OPTIONAL_HEADER, the IMAGE_DIRECTORY_ENTRY_SECURITY IMAGE_DATA_DIRECTORY, and the public key certificate table, which always resides at the end of the image file.

If the boot application is validated, then the root public key of the embedded public key certificate chain must match one of the hardcoded Microsoft’s root public keys. If the boot manager cannot validate the integrity of the boot application, then the boot manager does not continue to load Windows 7/WS08R2 modules, instead displaying an error message below along with the full name of the boot application that failed the integrity check.

After the boot application’s integrity has been determined, the boot manager attempts to load the boot application. When configured, the full volume encryption (FVE) facility within the Windows 7/WS08R2 boot manager also conducts its own independent SHA-256 hash based validation of the boot applications as identified above. If the boot application is successfully loaded, the boot manager then transfers execution to the loaded application.

After the Windows 7/WS08R2 Winload boot application is loaded, it receives the transfer of execution from the boot manager. During its execution, Winload attempts to load the Windows 7/WS08R2 kernel (ntoskrnl.exe) together with a number of critical drivers. Among the modules that Winload must validate in the Portable Executable (PE) image file format, are the cryptography related modules listed below. These modules are listed in a hardcoded list.

- The Windows 7/ WS08R2 kernel;
- The Windows 7/ WS08R2 kernel security device driver;
- The Windows 7/ WS08R2 code integrity library module; and
- The BitLocker™ drive encryption filter driver.

The four image files above have their trusted SHA hashes stored in catalog files that reside in the local machine catalog directory.

Because they are PKCS #7 SignedData messages, catalog files are 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 indicating that Microsoft is the publisher of the Windows 7/WS08R2 image files. These Microsoft's root public keys are hardcoded in the Winload boot application.

If the image files are validated, their SHA hashes, as calculated by the Winload boot application, must match their trusted SHA hashes in a Microsoft's catalog file, which has been verified by the Winload boot application. A SHA hash of an image file is calculated for the whole file, with the exception of the following three elements which are excluded from the hash calculation: the CheckSum field in the IMAGE_OPTIONAL_HEADER, the IMAGE_DIRECTORY_ENTRY_SECURITY IMAGE_DATA_DIRECTORY, and the public key certificate table, which always resides at the end of the image file.

Should the Winload boot application be unable to validate the integrity of one of the Windows 7/WS08R2 image files, the Winload boot application does not continue to load other Windows 7/WS08R2 image files. Rather it displays an error message, along with the full name of the Windows 7/WS08R2 image file which does not have the validated integrity.

In addition, Windows File Protection maintains a set of protected files that are stored in a cache along with cryptographic hashes of each of those files. Once the system is initialized, Windows File Protection is loaded and will scan the protected files to ensure they have valid cryptographic hashes. Windows File Protection also registers itself to be notified should any of the protected files be modified so that it can recheck the cryptographic checksum at any point while the system is operational. Should the any of the cryptographic hash checks fail, the applicable file will be restored from the cache.

SFR Mapping:

The **TSF Protection** function satisfies the following SFRs:

- FPT_ITT.1, FPT_ITT.3: The TSF provides internet-based standard protocols for IP security and Key management. IPSec with AH and ESP implementations protect transferred TSF data from disclosure and modification. AH provides data signature functionality to protect against modification; ESP provides encryption to protect against disclosure as well as modification. The TSF implements IP AH. AH provides integrity, authentication and anti-replay. AH uses a hashing algorithm, such as SHA, to compute a keyed message hash for each IP packet. Additionally, IPSec policies and filters may be configured to reject the packet or audit the event if the results of a service applied to a packet challenges the integrity of the packet (modification, insertion of data, replay of data). Any packets rejected as a result of an integrity error are rejected and the event is audited.

- FPT_RCV.1: The TSF enters a halted state upon failure and allows the user to restart the operating system in a limited operational mode where recovery can be attempted.
- FPT_WPF_EX.1: The TSF implements memory protection on 64-bit architectures by not executing code on pages marked for data only. The owing process has the ability to set the flags associated with its memory pages.
- FPT_WPF_EX.2: When configured, the TSF is capable of performing full disk volume encryption in order to protect the disk contents (TSF, TSF data, and user data) from potential modification and disclosure. Only when appropriate credentials are provided can the TSF be made to start or the contents of the disk be otherwise accessed.
- FPT_WPF_EX.3: The TSF is capable of encrypting the content of removable USB storage devices using BitLocker. Furthermore, the TSF provides the ability to require the use of this feature in order to write content to removable USB storage devices.
- FPT_WPF_EX.4: The TSF includes a Network Policy Service role that allows administrators to define network access requirements and also serves to compare reported health profiles against the requirements in order to decide which, if any, network services will be provided to an applicable client. The TSF also includes agents that serve as the counterpart for the health profiles based on TSF settings and installed applications. The agents report their health profiles to the server when network access is needed so that the server can provide applicable network access credentials (or not) based on comparing that data to its configured health requirements.
- FPT_WPF_EX.5: When a supporting TPM chip is present, the TSF can use it to store FVE encryption keys. When starting the TOE, the TPM chip will release the FVE encryption keys only when the integrity of selected operating system components can be confirmed.
- FPT_STM.1, FMT_MTD.1a (partial): The real-time clock in each Windows 7 and Windows Server 2008 R2 platform, in conjunction with periodic domain synchronization and restricting the ability to change the clock to authorized administrators, provides a reliable source of time stamps for the TSF.
- FPT_TRC_EXT.1: The TSF provides consistency of replicated GPOs and DS data by implementing a well-defined TSF replication algorithm that results in replication as soon as possible.
- FPT_TST_EXT.1: The TSF includes cryptographic self-tests in accordance with FIPS 140-2 per the FIPS certification (certificates #1319, #1321, #1326, #1327, #1328, #1329, #1330, #1331, #1332, #1333, #1334, #1335, #1336, #1337, #1338, and #1339). Beyond this, the TSF includes a number of checks performed during the boot sequence designed to ensure cryptographically that TSF images have not been modified. Windows File Protection also serves to help prevent modification or other corruption of TSF files. Finally, as described above, BitLocker enables Full

Volume Encryption for the operating system and all of its data. While BitLocker prevents disclosure of data, it also ensures the integrity of that data. If any changes are made, it will not decrypt properly resulting in a failure of the operating system to boot and run should any changes be made to the TOE or its data stores.

6.1.7 Resource Utilization Function

The TSF provides a function that can limit the amount of disk space that can be used by an identified user on a specific NTFS-formatted disk volume. Each NTFS volume has a set of properties, including a description of applicable disk quotas that can be changed only by an authorized administrator.[31] These properties allow an authorized administrator to enable or disable quota management on the selected volume, specify default and specific quota thresholds and warning levels, and select the action to take when quotas are exceeded.

The disk space quota threshold and warning level properties can be specified per user account, while each of the other properties apply to all users of the volume. Any disk space that is used is associated with the account that owns the object, based on the owner property of the object. When quota management is enabled, the first time that an object is created on a volume for a given account, a quota record will be created for that account (if it hasn't already been explicitly created). This quota record is initially assigned the default disk space and warning levels and is used subsequently to manage that account's use of disk space. Whenever a given account causes more disk space to be allocated, the quota record for that account is modified and the thresholds are checked. If the warning level or disk space quota is exceeded, the administrator-selected action is taken.

SFR Mapping:

The **Resource Utilization** function satisfies the following SFR:

- FRU_RSA.1: The quota feature of NTFS provides an authorized administrator the ability to effectively limit the total amount of disk space that a specified user can use on a specific NTFS disk volume.

6.1.8 Session Locking Function

The TSF provides the ability for a user to lock their interactive logon session immediately or after a user-defined time interval. Additionally, the TSF provides the ability for the administrator to specify a defined interval of inactivity after which the session will be locked. Once a user is logged on, they can invoke the session locking function by using the same key sequence used to invoke the trusted path (**Ctrl+Alt+Del**). This key sequence is captured by the TSF and cannot be intercepted or altered by any user process. The result of that key sequence is a menu of functions, one of which is to lock the workstation.

Alternately, a user can invoke a function to set screen saver properties for their interactive logon session. The user can select a program to use as a screen saver, the amount of inactivity before the

[31] Note that while NTFS can support POSIX file-naming conventions, the Windows POSIX subsystem is not delivered as part of the Windows DVD and not included within the scope of this evaluation.

screen saver will start, and whether a password will be required to resume the user's session (effectively making the screen saver a session lock). The TSF constantly monitors the mouse and keyboard for activity and if they are inactive for the user-specified time period, the TSF will lock the workstation (assuming the user configured it to lock the session) and execute the screen saver program (assuming the user selected a screen saver program). Note that if the workstation was not locked manually, the TSF will start the screen saver program if and when the inactivity period is exceeded.

When the workstation is locked manually, or when there is mouse or keyboard activity after the screen saver program has started (assuming a password is required, otherwise the session immediately resumes), the TSF will display the user's default background and a dialog indicating that the user must use the **Ctrl+Alt+Del** sequence to re-authenticate.

Regardless of how the workstation was locked, the user must use the **Ctrl+Alt+Del** function that will result in an authentication dialog. The user must then re-enter their password, which has been cached by the local system from the initial logon, after which the user's display will be restored and the session will resume. Alternately, an authorized administrator can enter their administrator identity and password in the authentication dialog. If the TSF can successfully authenticate the administrator, the user will be logged off, rather than returning to the user's session, leaving the workstation ready to authenticate a new user.

The web server (IIS) configuration values (in the metabase) includes a value that defines the time in seconds that IIS waits before it disconnects an inactive session. Only an authorized administrator can define this value.

SFR Mapping:

The **Session Locking** function satisfies the following SFR:

- FTA_SSL.1: Windows 7 and Windows Server 2008 R2 allows users and the authorized administrator to define an inactivity interval, after which their session will be locked. The locked display has only the user's default background, instructions to unlock, and optionally the output from a user-selected screen saver program. The user must re-enter their password to unlock the workstation.
- FTA_SSL.2: Windows 7 and Windows Server 2008 R2 also allows a user to directly invoke the session lock as described above.
- FTA_SSL.3: IIS disconnects an inactive session after the authorized administrator defined time has elapsed.
- FMT_MTD.1a (partial): The TSF allows an authorized user to define and modify the time interval of inactivity before the session associated with that user will be locked.

6.1.9 Certificate Services Function

The Certificate Services function covers a number of topics related to providing certificate services to users. Among those functions are

- Remote Certificate Request Data Entry and Certificate Status Export
- Certificate Services-related Security Management
- Key Management.

6.1.9.1 Remote Certificate Request Data Entry & Certificate and Certificate Status Export

The Windows Certificate Services server role processes certificate requests formatted according to the following standards which, in conjunction with the Identification security function and I&A performed in the TOE, provide the verification of origin framework for the TOE to follow:

- PKCS #7 (Cryptographic Message Syntax Standard)
- PKCS #10 (Certification Request Syntax Standard)
- RFC 2797 CMC (Certificate Management Messages over Cryptographic Message Syntax)

The Windows Certificate Services role generates certificates and certificate revocation lists according to the following standard which provides a verification of origin framework for users of certificates and CRLs to follow:

- RFC 3280 Internet X.509 PKI Certificate and CRL Profile (which is consistent with ITU-T Recommendation X.509.

In servicing certification requests or renewal of certificates, a Windows Enterprise Certificate Authority (CA) ensures that the certificate request is digitally signed and that the caller is the subject of the certificate request. The Windows Certificate Services role will not accept a certificate request or certificate renewal request if it is not signed. Furthermore, the Windows Certificate Services Enterprise CA will not issue a certificate if the user submitting the request is different from the certificate subject specified in the request.

6.1.9.2 Certificate Management

The Windows Certificate Services role provides the ability to issue certificates, publish CRLs, and generate OCSP responses. Certificate templates (or profiles) are stored and managed securely by the Active Directory. Templates contain attributes and information that may be included in the request or will be automatically used in the request if it is not present in the request.

For Enterprise CAs, every certificate request is based on a template. If it is not based on a template, the certificate request will be rejected. During the certificate request, the Certificate Service validates that all required attributes are provided or the certificate request will be denied.

The Windows Certificate Services role allows for qualified subordination which can place certificate issuance constraints on subordinate CAs and can place usage constraints on the certificates they issue. With qualified subordination, a subordinate CAs can be focused according to specific certification needs allowing for more efficient administration. Qualified subordination also allows for the establishment of trust between CAs in separate trust hierarchies. This type of trust relationship is also called cross-certification. With this trust relationship, qualified subordination is not limited to subordinate CAs.

Trust between hierarchies may be established using a subordinate CA in one hierarchy and the root CA in another hierarchy.

Qualified subordination extends the trust hierarchy by allowing the ability to place additional trust conditions within and between the namespaces managed by the PKI. With qualified subordination, the qualified subordinate CAs in the trust hierarchy can each have different rules governing how they will issue certificates and how their certificates may be used. All constraints that are placed on a qualified subordinate CA are defined when the cross-certificate template was created. [32]

The Windows Certificate Services role publishes CRLs that identifies which certificates in the certificate database that have been revoked. The Windows Certificate Services role can publish two types of CRLs: Base CRLs and Delta CRLs. A Base CRL identifies all the certificates that have been revoked and a Delta CRLs identifies the certificates that have been revoked since the last published Base CRL. The Windows Certificate Services role can publish CRLs automatically based upon a configured time period or upon the manual invocation by the CA Administrator.

The Windows Certificate Services role can also be configured to generate OCSP responses in accordance with IETF RFC 2560 with some mandatory field values as indicated in the rationale below.

6.1.9.3 Key Management

The key management function is concerned with the management of keys, such as private keys, that are used to support security functions and the public keys associated with the certificates provided to users. Windows protects the certificates with digital signatures to ensure the integrity certificate-related information.

Windows relies on FIPS 140-2 validated cryptographic security modules for key generation (when Windows creates, i.e., does not import a key) for certificates, key storage and key destruction through zeroization.

6.1.9.4 Certificate Services-related Security Management

The following table defines restrictions associated with managing Certificate Services provided by the Windows Certificate Services role:

Function	Authorized Role
Certificate Registration	The capability to approve fields or extensions to be included in a certificate shall be restricted to Officers.
[Certificate] Data Export and	The export of CIMC private keys shall require the

[32] See http://technet.microsoft.com/en-us/library/cc739804(WS.10).aspx

Function	Authorized Role
Output	authorization of at least two Administrators or one Administrator and one Officer, Auditor, or Operator.
Certificate Status Change Approval	Only Officers shall configure the process used to approve the revocation of a certificate or information about the revocation of a certificate. Only Officers shall configure the process used to approve the placing of a certificate on hold or information about the on hold status of a certificate.
Certificate Issuing Management Component (CIMC) Configuration	The capability to configure any TSF functionality shall be restricted to Administrators. (This requirement applies to all configuration parameters unless the ability to configure that aspect of the TSF functionality has been assigned to a different role elsewhere in this document.)
Certificate Profile Management	The capability to modify the certificate profile shall be restricted to Administrators.[33]
Revocation Profile Management	The capability to modify the revocation profile shall be restricted to Administrators.
Certificate Revocation List Profile Management	The capability to modify the certificate revocation list profile shall be restricted to Administrators.[34]
Online Certificate Status Protocol (OCSP) Profile Management	The capability to modify the OCSP profile shall be restricted to Administrators[35]

[33] i.e., FMT_MOF_CIMC.3 Extended certificate profile management

[34] i.e., FMT_MOF_CIMC.5 Extended certificate revocation list profile management

[35] i.e., FMT_MOF_CIMC.6 OCSP profile management

Windows uses the registry to store certificates (i.e., the registry is the certificate database). [36]

The **Certificate Services** function satisfies the following SFRs:

- FCO_NRO_CIMC.3: The Windows Certificate Services role generates certificates and CRLs according to the RFC 3280 standard which provides a verification of origin framework. Therefore, Windows provides the ability to prove the origin of status information it generates. Additionally, Windows uses DCOM authentication based interfaces to communicate with the Certificate Services role. When certificate requests are received the identity of the requesting user is impersonated and the request is completed in the context of that user.

  For new enrollments, depending upon template configuration, the information pertaining to the subject of the certificate is typically retrieved from the Active Directory user account object of the authenticated user submitting the request. In addition, when renewing a certificate, the request is parsed and the data is analyzed to ensure that the certificate subject name matches the authenticated user that submitted the request.

  The Windows Certificate Services role processes certificate requests formatted according to the PKCS #7, PKCS #10, and RFC 2797 standards. The Windows Certificate Services role can be configured to use the rest of the RDN prefix of the submitted request. Alternatively, the Windows Certificate Services role can be configured to obtain the full subject DN from the Active Directory using the authenticated subject identity.

  When certificate revocation requests are received, the role/authorization of the requesting user is verified. The TOE Identification security function and I&A provide the verification of origin of revocation request.

- FCO_NRO_CIMC.4: Certificate requests made by the Certificates MMC Snap-in or CERTREQ.EXE command line tool must be encoded using PKCS #10 or CMC formats. These formats inherently support the request being signed using the private key corresponding to the public key in the certificate request. This provides proof of possession of the private key. The Windows Certificate Services role does not accept any other security relevant information outside of certificate requests. When certificate requests are received by an enrollment proxy, the identity of the requesting user is impersonated and the request is completed in the context of that user. Additionally, the request is parsed to ensure that the certificate subject name matches the user who was authenticated and submitted the request. For certificate renewal, the subject must send a PKCS#7 request signed using a current valid signature key.
- FDP_ACF_CIMC.2: The Certificate Authority private key is stored by Windows in the hardware Cryptographic Service Module (HSM), when present, or encrypted using the TOE’s FIPS certified software cryptographic module. No other private keys are collected or stored.

[36] Note that when Windows stores a private key that is not embedded in a certificate, the keys are protected by the Data Protection API and stored within NTFS volumes. See the *Certificates and Private Keys* section at http://support.microsoft.com/kb/309408 for additional information.

- FDP_ACF_CIMC.3: Except when configured for key archival, the Windows Certificate Services role does not collect or store user secret keys.
- FDP_CIMC_CER.1: As described above, the Windows Certificate Services role verifies that the prospective certificate subject possesses the private key that corresponds to the public key in the certificate request before issuing a certificate, whenever the private key may be used to generate digital signatures. The Windows Certificate Services Enterprise CA provides standard templates for the certificates and ensures that certificates are consistent with the currently selected template and shall only generate certificates that comply with requirements for certificates as specified in ITU-T Recommendation X.509. At a minimum, the TSF shall ensure that:

  a) The version field shall contain the integer 0, 1, or 2.

  b) If the certificate contains an issuerUniqueID or subjectUniqueID then the version field shall contain the integer 1 or 2.

  c) If the certificate contains extensions then the version field shall contain the integer 2.

  d) The serialNumber shall be unique with respect to the issuing Certification Authority.

  e) The validity field shall specify a notBefore value that does not precede the current time and a notAfter value that does not precede the value specified in notBefore.

  f) If the issuer field contains a null Name (e.g., a sequence of zero relative distinguished names), then the certificate shall contain a critical issuerAltName extension.

  g) If the subject field contains a null Name (e.g., a sequence of zero relative distinguished names), then the certificate shall contain a critical subjectAltName extension.

  h) The signature field and the algorithm in the subjectPublicKeyInfo field shall contain the OID for a FIPS-approved or recommended algorithm.

- FDP_CIMC_CRL.1: The Windows Certificate Services role provides standard templates for the CRLs to ensure that CRLs are consistent. The TOE ensures that the following fields and extensions in any CRL issued contain values in accordance with RFC3280:

1. The version field shall contain the integer 2.
2. Note that the CRL does not contain any critical extensions.
3. The issuer field shall contain the issuing certificate authority’s distinguished name (DN) represented using an X.500 DN.
4. The signature and signatureAlgorithm fields shall contain the OID for a FIPS-approved digital signature algorithm.

5. The thisUpdate field shall indicate the issue date of the CRL. The time specified in the nextUpdate field (if populated) shall not precede the time specified in the thisUpdate field.
6. The CRL Number extension shall indicate a monotonically increasing sequence number for each CRL being issued.
7. By default, the authority key identifier extension shall contain a key hash as a means to identify the public key corresponding to the private key used to sign a CRL. As an option, the authority key identifier extension may instead contain a numeric representation of the issuer name and serial number from the CRL issuer's certificate as a means to identify the public key corresponding to the private key used to sign a CRL.
8. The freshest CRL extension shall contain the URLs to fetch the delta CRL.
9. There shall be a sequence of zero or more revoked certificates with the following fields represented for each revoked certificate.
    a. The certificate serial number field shall contain the serial number assigned by the issuing certificate authority for each revoked certificate.
    b. The revocation date field shall contain the date at which the revocation took place.
    c. The reason code field shall identify the reason for the certificate revocation, which may be Unspecified, KeyCompromise, CACompromise, AffiliationChanged, Superseded, CessationOfOperation, CertificateHold, and RemoveFromCRL.

- FDP_CIMC_CSE.1: The Windows Certificate Services role provides certificate status information by following means: CRLs (X.509/ RFC 3280): The Windows Certificate Services role provides the ability to configure the specific details of the CRLs for each CA to the Certificate Administrator. However, the system enforces compliance with X.509 by limiting the configurable options. The CRLs will always contain the RFC-required fields: Signature Algorithm identifier, issuer Name, this Update Date, Revoked Certificate and a Signature. The format of the exported CRL conforms to the X.509 standard for CRLs specified in RFC 3280, except that the critical Issuing Distribution extension is not asserted in specific circumstances when the CRL does not cover certificates where the CA key signing the certificates is different from the CA key signing the CRL.
- FDP_CIMC_OCSP.1: The Windows Certificate Services role generates OCSP responses in accordance with IETF RFC 2560 with the following minimum field values:

    1. The version field shall contain a 0.

    2. If the issuer field contains a null Name (e.g., a sequence of zero relative distinguished names), then the response shall contain a critical issuerAltName extension.

    3. The signatureAlgorithm field shall contain the OID for a FIPS-approved digital signature algorithm.

    4. The thisUpdate field shall indicate the time at which the status being indicated is known to be correct.

5. The producedAt field shall indicate the time at which the OCSP responder signed the response.

6. The time specified in the nextUpdate field (if populated) shall not precede the time specified in the thisUpdate field.

- FDP_ETC_CIMC.5: The Windows Certificate Services role does not support the export of user private or secret keys. The Certificate Services may be configured to store the Certificate Authority private key in a hardware cryptographic service module. [37]
- FMT_MOF.1c: Windows restricts the ability to modify management functionality to a specific role. Windows ensures the user is a member of the appropriate role before the management behavior is modified. Restrictions for specific functions are as presented in the table above.
- FMT_MOF_CIMC.3: The Windows Certificate Services Enterprise CA provides standard templates for certificates and ensures that certificates it creates are consistent with the currently selected template. These templates conform to the X.509 standard. Certificate templates are stored as directory objects in the Active Directory. A default set of X.509-compliant templates is assigned to each Certificate Server when it is created, and the template selection may be modified by the Certificate Administrator through the Certificate Template MMC snap-in. Certificates issued by a Certificate Server must conform to one of its assigned templates. Windows provides the ability to configure the specific details of the certificates (i.e., Domain Name (DN) Attributes or Extensions) for each Certificate Server to the Administrator. The Certificate Administrator can specify acceptable values for the following fields and extensions: key owner's identifier; the algorithm identifier for the subject's public/private key pair; the identifier of the certificate issuer; the length of time for which the certificate is valid; keyUsage; basicContraints; certificatePolicies, and certificate extensions.
- FMT_MOF_CIMC.5: The Windows Certificate Services role generates CRLs according to a hardcoded template, to ensure CRLs are always consistent with the standard certificate revocation list template. The value of the Issuer field is determined by the name of the issuing Certificate Server and the value of the nextUpdate field is controlled by the Certificate Administrator. The configurable values are stored in the registry. Windows does not support the issuerAltName field within the CRL.
- FMT_MOF_CIMC.6: The Windows Certificate Services role generates OSCP responses according to a hardcoded template. The configurable values are stored in the registry.
- FMT_MTD_CIMC.4, FMT_MTD_CIMC.5, FMT_MTD_CIMC.7: The CA private key may be stored in the HSM, when present, or in a form encrypted by the FIPS-certified software cryptographic module in the TOE. All encryption is performed using FIPS 140-2 validated cryptographic modules. The only private key that is stored is CA private key, unless the administrator chooses to store archived keys.

[37] Please note that this functional requirement applies to the Certificate Services role. Keys and tokens used by features like the Encrypting File System and BitLocker may export protected keys to a smartcard.

7 Protection Profile Claims

This section provides the PP conformance claim statements and supporting justifications of conformance with the US Government Protection Profile for General-Purpose Operating Systems in a Networked environment, version 1.0, 30 August 2010 (GPOSPP).

7.1 Security Problem Definition

The statements of threats, policies, and assumptions have been copied verbatim from the GPOSPP with the exception of the addition of T.REPLAY. T.REPLAY was in a previous version of the GPOSPP and is retained in this Security Target since that threat is countered by the TOE.

7.2 Security Objectives

The statements of objectives for the TOE and its operational environment have been copied verbatim from the GPOSPP.

7.3 Security Requirements

The statements of security functional requirements (SFRs) for the TOE have been copied verbatim (with some minor label changes to match the conventions used in this ST to reflect iterations) from the GPOSPP. SFR operations left incomplete in the GPOSPP have been completed in this Security Target (ST) as identified in section **5.2 TOE Security Functional Requirements**. Additional SFRs have been copied verbatim from the CC (with operations performed as appropriate), have been defined in section **5.1**, or have been borrowed from the (draft) *Certificate Issuing and Management Components For Basic Robustness Environments Protection Profile, Version 1.0, April 27, 2009* (CIMCPP). The following table identifies the SFRs and their source.

Table 7-1 Security Functional Requirement Source

Requirement Component	Requirement Source and Operations
FAU_GEN.1: Audit data generation	GPOSPP – Refined
FAU_GEN.2: User identity association	GPOSPP – No operation
FAU_SAR.1: Audit review	GPOSPP – No operation
FAU_SAR.2: Restricted audit review	GPOSPP – No operation
FAU_SAR.3: Selectable audit review	GPOSPP – Refined
FAU_SEL.1: Selective audit	GPOSPP – No operation
FAU_STG.1: Protected audit trail storage	GPOSPP – No operation
FAU_STG.3: Action in case of possible audit data loss	GPOSPP – No operation
FAU_STG.4: Prevention of Audit Data Loss	CC – Selection and assignment
FCO_NRO_CIMC.3: Enforced proof of origin and verification of origin	CIMCPP - Assignment and refinement
FCO_NRO_CIMC.4: Advanced verification of origin	CIMCPP – No operation
FCS_BCM_EXT.1: Baseline Cryptographic Module	GPOSPP – No operation
FCS_CKM.1a: Cryptographic Key Generation (for	GPOSPP – Assignment

Requirement Component	Requirement Source and Operations
symmetric keys)	
FCS_CKM.1b: Cryptographic Key Generation (for asymmetric keys)	GPOSPP – Selection, assignment, and refinement
FCS_CKM.4: Cryptographic key destruction	GPOSPP – No operation
FCS_COA_EXT.1: Cryptographic Operations Availability	GPOSPP – Assignment
FCS_COP.1a: Cryptographic Operation (for data encryption/decryption)	GPOSPP – Assignment and selection
FCS_COP.1b: Cryptographic Operation (for cryptographic signature)	GPOSPP – Selection and assignment
FCS_COP.1c: Cryptographic Operation (for cryptographic hashing)	GPOSPP – Selections
FCS_COP.1d: Cryptographic Operation (for data encryption/decryption)	CC – Assignments
FCS_COP.1e: Cryptographic Operation (for cryptographic signature)	CC – Assignments
FCS_COP.1f: Cryptographic Operation (ECDH Key Agreement)	CC – Assignments
FCS_COP.1g: Cryptographic Operation (ECDSA Key Agreement)	CC – Assignments
FCS_RBG_EXT.1: Random Number Generation	GPOSPP – Selections
FDP_ACC.2a: Complete access control	GPOSPP – Augmentation from FDP_ACC.1
FDP_ACC.2b: WEBUSER (WU) Complete Access Control	CC – Assignments
FDP_ACC.2c: Content-Provider (CP) Complete Access Control	CC – Assignments
FDP_ACC.2d: Mandatory Integrity Control Policy	CC – Assignments
FDP_ACF.1a: Security attribute based access control	GPOSPP – Assignments and selection
FDP_ACF.1b: WEBUSER Access Control Functions	CC – Assignments
FDP_ACF.1c: Content Provider Access Control Functions	CC – Assignments
FDP_ACF.1d: Mandatory Integrity Control Functions	CC – Assignments
FDP_ACF_CIMC.2: User private key confidentiality protection	CIMCPP – No operation
FDP_ACF_CIMC.3: User secret key confidentiality protection	CIMCPP – No operation
FDP_CIMC_CER.1: Certificate Generation	CIMCPP – Assignment
FDP_CIMC_CRL.1: Certificate revocation list validation	CIMCPP – No operation
FDP_CIMC_CSE.1: Certificate status export	CIMCPP – Assignment
FDP_CIMC_OCSP.1: OCSP basic response validation	CIMCPP – No operation
FDP_ETC_CIMC.5: Extended user private and secret key export	CIMCPP – No operation
FDP_IFC.1a: IPSec Subset Information Flow Control	CC – Assignments
FDP_IFC.1b: Windows Firewall Connection Subset Information Flow Control	CC – Assignments
FDP_IFF.1a: IPSec Simple Security Attributes	CC – Assignments
FDP_IFF.1b: Windows Firewall Connection Simple	CC – Assignments

Requirement Component	Requirement Source and Operations
Security Attributes	
FDP_ITT.1: Basic Internal Transfer Protection	CC – Assignment and selection
FDP_RIP.2: Full residual information protection	GPOSPP – Selection
FDP_UCT.1: WEBUSER Basic Data Exchange Confidentiality	CC – Assignment and selection
FDP_UIT.1: WEBUSER SFP Data Exchange Integrity	CC – Assignment and selections
FIA_AFL_EXT.1: Authentication Failure Handling	GPOSPP – No operation
FIA_ATD.1: User attribute definition	GPOSPP – Assignments
FIA_SOS.1: Verification of secrets	GPOSPP – No operation
FIA_UAU.1: Timing of authentication	GPOSPP – No operation
FIA_UAU.6: Re-authenticating	GPOSPP – No operation
FIA_UAU.7: Protected authentication feedback	GPOSPP – No operation
FIA_UID.1: Timing of identification	GPOSPP – No operation
FIA_USB.1: User-subject binding	GPOSPP – Assignment and refinement
FMT_MOF.1a: Management of Security Functions Behavior (for specification of auditable events)	GPOSPP – No operation
FMT_MOF.1b: Management of Security Functions Behavior (for authentication data)	GPOSPP – No operation
FMT_MOF.1c: Management of security functions behavior (certificate services)	CIMCPP - Refinement
FMT_MOF_CIMC.3: Extended certificate profile management	CIMCPP – No operation
FMT_MOF_CIMC.5: Extended certificate revocation list profile management	CIMCPP – No operation
FMT_MOF_CIMC.6: OCSP profile management	CIMCPP – No operation
FMT_MSA.1a: Management of Security Attributes (for Discretionary Access Control)	GPOSPP – Assignment
FMT_MSA.1b: Management of Security Attributes (for Object Ownership)	GPOSPP – No operation
FMT_MSA.1c : Management of IPSec Object Security Attributes	CC – Assignments and selection
FMT_MSA.1d : Management of Windows Firewall Connection Object Security Attributes	CC – Assignments and selection
FMT_MSA.1e : Management of WEBUSER Object Security Attributes	CC – Assignments and selection
FMT_MSA.1f : Management of CONTENT-PROVIDER Object Security Attributes	CC – Assignments and selection
FMT_MSA.1g : Management of Mandatory Integrity Control Security Attributes	CC – Assignments and selection
FMT_MSA.2: Secure security attributes	GPOSPP – No operation
FMT_MSA.3a: Static attribute initialization	GPOSPP – No operation
FMT_MSA.3b: IPSec Static Attribute Initialization	CC – Assignments and selection
FMT_MSA.3c: Windows Firewall Connection Static Attribute Initialization	CC – Assignments and selection

Requirement Component	Requirement Source and Operations
FMT_MSA.3d: WEBUSER Static Attribute Initialization	CC – Assignments and selection
FMT_MSA.3e: CONTENT-PROVIDER Static Attribute Initialization	CC – Assignments and selection
FMT_MSA.3f: Mandatory Integrity Attribute Initialization	CC – Assignments and selection
FMT_MTD.1a: Management of TSF Data (for general TSF data)	GPOSPP – No operation
FMT_MTD.1b: Management of TSF Data (for audit data)	GPOSPP – No operation
FMT_MTD.1c: Management of TSF Data (for initialization of user security attributes)	GPOSPP – No operation
FMT_MTD.1d: Management of TSF Data (for modification of user security attributes, other than authentication data)	GPOSPP – No operation
FMT_MTD.1e: Management of TSF Data (for modification of authentication data)	GPOSPP – No operation
FMT_MTD.1f: Management of TSF Data (for reading of authentication data)	GPOSPP – No operation
FMT_MTD.1g: Management of TSF Data (for critical cryptographic security parameters)	GPOSPP – No operation
FMT_MTD_CIMC.4: TSF private key confidentiality protection	CIMCPP – No operation
FMT_MTD_CIMC.5: TSF secret key confidentiality protection	CIMCPP – No operation
FMT_MTD_CIMC.7: Extended TSF private and secret key export	CIMCPP – No operation
FMT_REV.1a: Revocation (to authorized administrators)	GPOSPP – No operation
FMT_REV.1b: Revocation (to owners and authorized administrators)	GPOSPP – No operation
FMT_SAE.1: Time-limited authorization	GPOSPP – No operation
FMT_SMF.1: Specification of Management Functions	GPOSPP – No operation
FMT_SMR.1: Security roles	GPOSPP – Assignment
FPT_ITT.1: Basic internal TSF data transfer protection	GPOSPP – Assignment
FPT_ITT.3: TSF data integrity monitoring	GPOSPP – Assignments
FPT_RCV.1: Manual recovery	GPOSPP – No operation
FPT_WPF_EX.1: TSF Hardware Protection	ST (extended)
FPT_WPF_EX.2: TSF Disk Volume Protection	ST (extended)
FPT_WPF_EX.3: Removable USB Storage Device Protection	ST (extended)
FPT_WPF_EX.4: Network Access Protection	ST (extended)
FPT_WPF_EX.5: TPM Full Volume Encryption Support	ST (extended)
FPT_STM.1: Reliable time stamps	GPOSPP – No operation
FPT_TRC_EXT.1: Internal TSF Data Consistency	GPOSPP – No operation
FPT_TST_EXT.1: TSF Testing	GPOSPP – Assignment

Requirement Component	Requirement Source and Operations
FRU_RSA.1: Maximum quotas	GPOSPP – No operation
FTA_MCS.1: Basic limitation on multiple concurrent sessions	GPOSPP – No operation
FTA_SSL.1: TSF-initiated session locking	GPOSPP – No operation
FTA_SSL.2: User-initiated locking	GPOSPP – No operation
FTA_SSL.3: WEBUSER TSF-Initiated Termination	CC – Refinement and assignment
FTA_TAB.1: Default TOE access banners	GPOSPP – No operation
FTA_TAH.1: TOE access history	GPOSPP – No operation
FTA_TSE.1: TOE Session Establishment	CC – Assignment
FTP_TRP.1: Trusted Path	CC – Selections and assignment

The statement of security assurance requirements (SARs) is EAL4 augmented with ALC_FLR.3 which exceeds the EAL2 augmented with ALC_FLR.2 requirement by the GPOSPP. The increase in assurance requirements is predicated upon customer requirements to have more assurance in the security functions of products being deployed in security solutions.

7.4 Rationale

For all the content copied verbatim from the GPOSPP, the corresponding rationale in that PP remains applicable as presented therein. The additional SFRs and SARs in this ST serve only to enhance the assurance in the claimed functions and to extend the set of security functions offered by the TOE in a manner that does not conflict with any of the requirements in the GPOSPP. Rationale for the additional requirements is provided in the following section.

8 Rationale

This section provides the rationale for completeness and consistency of the ST. The rationale addresses the following areas:

- Security Objectives
- Security Functional Requirements
- TOE Summary Specification

8.1 Security Objectives Rationale

The security problem definition and all of the security objectives in this ST have been drawn from the GPOSPP, with the exception of T.REPLAY. T.REPLAY is addressed via O.TSF_CRYPTOGRAPHIC_INTEGRITY which is mapped to FPT_ITT.3 in the GPOSPP. The mechanisms used to address FPT_ITT.3 ensure that any replay of security relevant data between distributed parts of the TOE are protected from replay in addition to ensuring confidentiality and integrity.

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the requirements in the ST. Table 8-3 shows that the SFRs identified in this ST beyond those defined in the GPOSPP correspond to and support identified security objectives.

8.2.1 Security Functional Requirements Rationale

The following table represents a subset of the correspondence between the TOE SFRs and TOE security objectives. In particular the SFRs beyond those defined in the GPOSPP are addressed since the GPOSPP already provides correspondence for the SFRs found therein.

Table 8-1 Requirement to Security Objective Correspondence

Requirement	O.ACCESS	O.AUDIT_PROTECTION	O.CRYPTOGRAPHIC_SERVICES	O.DISCRETIONARY_ACCESS	O.PROTECT
FAU_STG.4		X			
FCO_NRO_CIMC.3					X
FCO_NRO_CIMC.4					X
FCS_COP.1d			X		

Requirement	O.ACCESS	O.AUDIT_PROTECTION	O.CRYPTOGRAPHIC_SERVICES	O.DISCRETIONARY_ACCESS	O.PROTECT
FCS_COP.1e			X		
FCS_COP.1f			X		
FCS_COP.1g			X		
FDP_ACC.2b				X	
FDP_ACC.2c				X	
FDP_ACC.2d					X
FDP_ACF.1b				X	
FDP_ACF.1c				X	
FDP_ACF.1d					X
FDP_ACF_CIMC.2					X
FDP_ACF_CIMC.3					X
FDP_CIMC_CER.1					X
FDP_CIMC_CRL.1					X
FDP_CIMC_CSE.1					X
FDP_CIMC_OCSP.1					X
FDP_ETC_CIMC.5					X
FDP_IFC.1a					X
FDP_IFC.1b					X
FDP_IFF.1a					X
FDP_IFF.1b					X
FDP_ITT.1					X
FDP_UCT.1					X
FDP_UIT.1					X
FMT_MOF.1c					X
FMT_MOF_CIMC.3					X
FMT_MOF_CIMC.5					X
FMT_MOF_CIMC.6					X
FMT_MSA.1c					X
FMT_MSA.1d					X
FMT_MSA.1e				X	
FMT_MSA.1f				X	
FMT_MSA.1g					X
FMT_MSA.3b					X
FMT_MSA.3c					X
FMT_MSA.3d				X	

Requirement	O.ACCESS	O.AUDIT_PROTECTION	O.CRYPTOGRAPHIC_SERVICES	O.DISCRETIONARY_ACCESS	O.PROTECT
FMT_MSA.3e				X	
FMT_MSA.3f					X
FMT_MTD_CIMC.4					X
FMT_MTD_CIMC.5					X
FMT_MTD_CIMC.7					X
FPT_WPF_EX.1					X
FPT_WPF_EX.2					X
FPT_WPF_EX.3					X
FPT_WPF_EX.4					X
FPT_WPF_EX.5					X
FTA_SSL.3	X				
FTA_TSE.1	X				
FTP_TRP.1	X				

O.ACCESS

The TOE will ensure that users gain only authorized access to it and to resources that it controls.

This objective is further supported by the following SFRs:

- FTA_SSL.3 serves to support ensuring that users gain only authorized access by terminating inactive sessions.
- FTA_TSE.1 serves to support ensuring that users gain only authorized access by limiting sessions based on expired authentication data, location, time, and day.
- FTA_TRP.1 serves to support ensuring that users gain only authorized by using trusted paths that serve to protect authentication sessions from disclosure.

O.AUDIT_PROTECTION

The TOE will provide the capability to protect audit information.

This objective is further supported by the following SFRs:

- FAU_STG.4 serves to help protect audit data by generating an alarm and taking a prescribed action when the available audit space is exhausted.

O.CRYPTOGRAPHIC_SERVICES

The TOE will make encryption services available to authorized users and/or user applications.

This objective is further supported by the following SFRs:

- FCS_COP.1d, FCS_COP.1e, FCS_COP.1f, and FCS_COP.1g serve to require additional cryptographic operations not already required in the GPOSPP.

O.DISCRETIONARY_ACCESS

The TOE will control access to resources based upon the identity of users and groups of users.

This objective is further supported by the following SFRs:

- FDP_ACC.2b, FDP_ACC.2c, FDP_ACF.1b, and FDP_ACF.1c serve to impose requirements for controlling access to web based user data/content.
- FMT_MSA.1e, FMT_MSA.1f, FMT_MSA.3d, and FMT_MSA.3e serve as the corresponding default access and security management requirements for the web based user data/content policies.

O.PROTECT

The TOE will provide mechanisms to protect user data and resources.

This objective is further supported by the following SFRs:

- FDP_ACC.2d, FDP_ACF.1d, FMT_MSA.1g, and FMT_MSA.3f serve to require the enforcement and management of a mandatory integrity policies designed to protect both users and the TSF.
- FDP_IFC.1a, FDP_IFC.1b, FDP_IFF.1a, FDP_IFF.1b, FMT_MSA.1c, FMT_MSA.1d, FMT_MSA.3b, and FMT_MSA.3c serve to require the enforcement and management of information flow type requirements designed to control access to available services in order to help protect the TSF (and users).
- FDP_ITT.1, FDP_UCT.1, and FDP_UIT.1 serve to require protection of network traffic from disclosure and modification, protection of users and potentially the TSF.
- FPT_WPF_EX.1 serves to protect some applications from being abused as a result of some common programming errors that result in the execution of data not intended to be executed.
- FPT_WPF_EX.2 and FPT_WPF_EX.3 serve to protect the TSF and user data on fixed and removable USB storage devices by means of encryption.
- FPT_WPF_EX.4 serves to protect the TSF and users by limiting access to network resources based on perceived health of would-be network clients.
- FPT_WPF_EX.5 serves to protect the TSF by storing FVE keys in a manner that offers additional protection in the event of inappropriate physical access to the TOE.

- FCO_NRO _CIMC.3, FCO_NRO_CIMC.4, FDP_ACF_CIMC.2, FDP_ACF_CIMC.3, FDP_CIMC_CER.1, FDP_CIMC_CRL.1, FDP_CIMC_CSE.1, FDP_CIMC_OCSP.1, FDP_ETC_CIMC.5, FMT_MOF.1c, FMT_MOF_CIMC.3, FMT_MOF_CIMC.5, FMT_MOF_CIMC.6, FMT_MTD_CIMC.4, FMT_MTD_CIMC.5, and FMT_MTD_CIMC.7 collectively ensure that the TOE can issue and protect cryptographic certificates and related information and also to ensure those capabilities can be appropriately managed by authorized administrators.

8.2.2 Requirement Dependency Rationale

The requirements in the GPOSPP are assumed to represent a complete set of requirements that serve to address any interdependencies. Given that all of the SFRs have been copied verbatim into this ST, those requirements are not addressed here. Similarly, it is assumed that EAL4 also serves to address any of its own interdependencies since it is a predefined package of the CC. As such, the following table addresses only the SFRs and SARs beyond those identified in the GPOSPP and EAL4. As can be seen in the table all dependencies are fulfilled.

Table 8-2 Security Requirement Dependencies

ST Requirement	CC Dependencies	ST Dependencies
FAU_STG.4	FAU_STG.1	FAU_STG.1
FCO_NRO_CIMC.3	FIA_UID.1	FIA_UID.1
FCO_NRO_CIMC.4	FCO_NRO_CIMC.3	FCO_NRO_CIMC.3
FCS_COP.1d	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1a and FCS_CKM.4
FCS_COP.1e	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1b and FCS_CKM.4
FCS_COP.1f	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1b and FCS_CKM.4
FCS_COP.1g	(FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1) and FCS_CKM.4	FCS_CKM.1b and FCS_CKM.4
FDP_ACC.2b	FDP_ACF.1	FDP_ACF.1b
FDP_ACC.2c	FDP_ACF.1	FDP_ACF.1c
FDP_ACC.2d	FDP_ACF.1	FDP_ACF.1d
FDP_ACF.1b	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.2b and FMT_MSA.3d
FDP_ACF.1c	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.2c and FMT_MSA.3e
FDP_ACF.1d	FDP_ACC.1 and FMT_MSA.3	FDP_ACC.2d and FMT_MSA.3f
FDP_ACF_CIMC.2	none	none
FDP_ACF_CIMC.3	none	none
FDP_CIMC_CER.1	none	none
FDP_CIMC_CRL.1	none	none
FDP_CIMC_CSE.1	none	none
FDP_CIMC_OCSP.1	none	none
FDP_ETC_CIMC.5	none	none
FDP_IFC.1a	FDP_IFF.1	FDP_IFF.1a
FDP_IFC.1b	FDP_IFF.1	FDP_IFF.1b
FDP_IFF.1a	FDP_IFC.1 and FMT_MSA.3	FDP_IFC.1a and FMT_MSA.3b

ST Requirement	CC Dependencies	ST Dependencies
FDP_IFF.1b	FDP_IFC.1 and FMT_MSA.3	FDP_IFC.1b and FMT_MSA.3c
FDP_ITT.1	(FDP_ACC.1 or FDP_IFC.1)	FDP_IFC.1a
FDP_UCT.1	(FTP_ITC.1 or FTP_TRP.1) and (FDP_ACC.1 or FDP_IFC.1)	FTP_TRP.1 and FDP_ACC.2b
FDP_UIT.1	(FTP_ITC.1 or FTP_TRP.1) and (FDP_ACC.1 or FDP_IFC.1)	FTP_TRP.1 and FDP_ACC.2b
FMT_MOF.1c	FMT_SMR.1 and FMT_SMF.1	FMT_SMR.1 and FMT_SMF.1
FMT_MOF_CIMC.3	FMT_SMR.1 and FMT_MOF.1	FMT_SMR.1 and FMT_MOF.1c
FMT_MOF_CIMC.5	FMT_SMR.1 and FMT_MOF.1	FMT_SMR.1 and FMT_MOF.1c
FMT_MOF_CIMC.6	FMT_SMR.1 and FMT_MOF.1	FMT_SMR.1 and FMT_MOF.1c
FMT_MSA.1c	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_IFC.1a
FMT_MSA.1d	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_IFC.1b
FMT_MSA.1e	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_ACC.2b
FMT_MSA.1f	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_ACC.2c
FMT_MSA.1g	FMT_SMR.1 and FMT_SMF.1 and (FDP_ACC.1 or FDP_IFC.1)	FMT_SMR.1 and FMT_SMF.1 and FDP_ACC.2d
FMT_MSA.3b	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1c and FMT_SMR.1
FMT_MSA.3c	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1d and FMT_SMR.1
FMT_MSA.3d	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1e and FMT_SMR.1
FMT_MSA.3e	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1f and FMT_SMR.1
FMT_MSA.3f	FMT_MSA.1 and FMT_SMR.1	FMT_MSA.1g and FMT_SMR.1
FMT_MTD_CIMC.4	none	none
FMT_MTD_CIMC.5	none	none
FMT_MTD_CIMC.7	none	none
FPT_WPF_EX.1	none	none
FPT_WPF_EX.2	none	none
FPT_WPF_EX.3	none	none
FPT_WPF_EX.4	none	none
FPT_WPF_EX.5	none	none
FTA_SSL.3	none	none
FTA_TSE.1	none	none
FTP_TRP.1	none	none
ALC_FLR.3	none	none

8.3 TSS Rationale

This Section, in conjunction with Section 6, the TSS, provides evidence that the SFs are suitable to meet the TOE security requirements.

Each subsection in the Section 6.1, TSFs, describes a SF of the TOE. Each description is followed with rationale that indicates which requirements are satisfied by aspects of the corresponding SF. The set of SFs work together to satisfy all of the SFRs. Furthermore, all of the SFs are necessary in order for the TSF to provide the required security functionality.

The collection of SFs work together to provide all of the security requirements as indicated in **Table 8-3**. The SFs described in the TSS and indicated in the tables below are all necessary for the required security functionality in the TSF.

Table 8-3 Requirement to Security Function Correspondence

Requirement	Audit	User Data Protection	Cryptographic Protection	I & A	Security Management	TSF Protection	Resource Utilization	Session Locking	Certificate Services Function
FAU_GEN.1	X								
FAU_GEN.2	X								
FAU_SAR.1	X								
FAU_SAR.2	X								
FAU_SAR.3	X								
FAU_SEL.1	X								
FAU_STG.1	X								
FAU_STG.3	X								
FAU_STG.4	X								
FCO_NRO_CIMC.3									X
FCO_NRO_CIMC.4									X
FCS_BCM_EXT.1			X						
FCS_CKM.1a			X						
FCS_CKM.1b			X						
FCS_CKM.4			X						
FCS_COA_EXT.1			X						
FCS_COP.1a			X						
FCS_COP.1b			X						
FCS_COP.1c			X						
FCS_COP.1d			X						
FCS_COP.1e			X						
FCS_COP.1f			X						
FCS_COP.1g			X						
FCS_RBG_EXT.1			X						
FDP_ACC.2a		X							
FDP_ACC.2b		X							

Requirement	Audit	User Data Protection	Cryptographic Protection	I & A	Security Management	TSF Protection	Resource Utilization	Session Locking	Certificate Services Function
FDP_ACC.2c		X							
FDP_ACC.2d		X							
FDP_ACF.1a		X							
FDP_ACF.1b		X							
FDP_ACF.1c		X							
FDP_ACF.1d		X							
FDP_ACF_CIMC.2									X
FDP_ACF_CIMC.3									X
FDP_CIMC_CER.1									X
FDP_CIMC_CRL.1									X
FDP_CIMC_CSE.1									X
FDP_CIMC_OCSP.1									X
FDP_ETC_CIMC.5									X
FDP_IFC.1a		X							
FDP_IFC.1b		X							
FDP_IFF.1a		X							
FDP_IFF.1b		X							
FDP_ITT.1		X							
FDP_RIP.2		X							
FDP_UCT.1		X							
FDP_UIT.1		X							
FIA_AFL_EXT.1				X					
FIA_ATD.1				X					
FIA_SOS.1				X					
FIA_UAU.1				X					
FIA_UAU.6				X					
FIA_UAU.7				X					
FIA_UID.1				X					
FIA_USB.1				X					
FMT_MOF.1a					X				
FMT_MOF.1b					X				
FMT_MOF.1c									X
FMT_MOF_CIMC.3									X
FMT_MOF_CIMC.5									X
FMT_MOF_CIMC.6									X
FMT_MSA.1a		X							
FMT_MSA.1b		X							

Requirement	Audit	User Data Protection	Cryptographic Protection	I & A	Security Management	TSF Protection	Resource Utilization	Session Locking	Certificate Services Function
FMT_MSA.1c		X							
FMT_MSA.1d		X							
FMT_MSA.1e		X							
FMT_MSA.1f		X							
FMT_MSA.1g		X							
FMT_MSA.2					X				
FMT_MSA.3 a		X							
FMT_MSA.3 b		X							
FMT_MSA.3 c		X							
FMT_MSA.3 d		X							
FMT_MSA.3 e		X							
FMT_MSA.3 f		X							
FMT_MTD.1a	X	X		X	X	X		X	
FMT_MTD.1b	X				X				
FMT_MTD.1c					X				
FMT_MTD.1d					X				
FMT_MTD.1e					X				
FMT_MTD.1f					X				
FMT_MTD.1g					X				
FMT_MTD_CIMC.4									X
FMT_MTD_CIMC.5									X
FMT_MTD_CIMC.7									X
FMT_REV.1a					X				
FMT_REV.1b		X							
FMT_SAE.1					X				
FMT_SMF.1					X				
FMT_SMR.1					X				
FPT_ITT.1						X			
FPT_ITT.3						X			
FPT_RCV.1						X			
FPT_WPF_EX.1						X			
FPT_WPF_EX.2						X			
FPT_WPF_EX.3						X			
FPT_WPF_EX.4						X			
FPT_WPF_EX.5						X			
FPT_STM.1						X			
FPT_TRC_EXT.1						X			

Requirement	Audit	User Data Protection	Cryptographic Protection	I & A	Security Management	TSF Protection	Resource Utilization	Session Locking	Certificate Services Function
FPT_TST_EXT.1						X			
FRU_RSA.1							X		
FTA_MCS.1				X					
FTA_SSL.1								X	
FTA_SSL.2								X	
FTA_SSL.3								X	
FTA_TAB.1				X					
FTA_TAH.1				X					
FTA_TSE.1				X					
FTP_TRP.1				X					

9 Appendix A — List of Abbreviations

Abbreviation	Meaning
3DES	Triple DES
ACE	Access Control Entry
ACL	Access Control List
ACP	Access Control Policy
AD	Active Directory
AES	Advanced Encryption Standard
AGD	Administrator Guidance Document
AH	Authentication Header
ALPC	Advanced Local Process Communication
ANSI	American National Standards Institute
API	Application Programming Interface
BTG	BitLocker To Go
CA	Certificate Authority
CBC	Cipher Block Chaining
CC	Common Criteria
CD-ROM	Compact Disk Read Only Memory
CIFS	Common Internet File System
CIMCPP	Certificate Issuing and Management Components For Basic Robustness Environments Protection Profile, Version 1.0, April 27, 2009
CM	Configuration Management; Control Management

Abbreviation	Meaning
COM	Component Object Model
CP	Content Provider
CPU	Central Processing Unit
CRL	Certificate Revocation List
CryptoAPI	Cryptographic API
CSP	Cryptographic Service Provider
DAC	Discretionary Access Control
DACL	Discretionary Access Control List
DC	Domain Controller
DEP	Data Execution Prevention
DES	Data Encryption Standard
DH	Diffie-Hellman
DHCP	Dynamic Host Configuration Protocol
DFS	Distributed File System
DNS	Domain Name System
DS	Directory Service
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
ECB	Electronic Code Book
EFS	Encrypting File System
ESP	Encapsulating Security Protocol
FEK	File Encryption Key
FIPS	Federal Information Processing Standard
FRS	File Replication Service
FSMO	Flexible Single Master Operation
FVE	Full Volume Encryption
GB	Gigabyte
GC	Global Catalog
GHz	Gigahertz
GPC	Group Policy Container
GPO	Group Policy Object
GPOSPP	US Government Protection Profile for General-Purpose Operating System in a Networked Environment
GPT	GUID Partition Table; Group Policy Template
GUI	Graphical User Interface
GUID	Globally Unique Identifiers
HTTP	Hypertext Transfer Protocol
HTTPS	Secure HTTP
I/O	Input / Output
I&A	Identification and Authentication
IA	Information Assurance
ICF	Internet Connection Firewall
ICMP	Internet Control Message Protocol

Abbreviation	Meaning
ICS	Internet Connection Sharing
ID	Identification
IETF	Internet Engineering Task Force
IFS	Installable File System
IIS	Internet Information Services
IKE	Internet Key Exchange
IP	Internet Protocol
IPv4	IP Version 4
IPv6	IP Version 6
IPC	Inter-process Communication
IPSec	IP Security
ISAPI	Internet Server API
IT	Information Technology
KDC	Key Distribution Center
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LPC	Local Procedure Call
LSA	Local Security Authority
LSASS	LSA Subsystem Service
LUA	Least-privilege User Account
MAC	Message Authentication Code
MB	Megabyte
MMC	Microsoft Management Console
NAC	(Cisco) Network Admission Control
NAP	Network Access Protection
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
NTFS	New Technology File System
NTLM	New Technology LAN Manager
OS	Operating System
PC/SC	Personal Computer/Smart Card
PIN	Personal Identification Number
PKCS	Public Key Certificate Standard
PKI	Public Key Infrastructure
PP	Protection Profile
RAID	Redundant Array of Independent Disks
RAM	Random Access Memory
RC4	Rivest's Cipher 4
RID	Relative Identifier
RNG	Random Number Generator
RPC	Remote Procedure Call
RSA	Rivest, Shamir and Adleman
RSASSA	RSA Signature Scheme with Appendix

Abbreviation	Meaning
SA	Security Association
SACL	System Access Control List
SAM	Security Assurance Measure
SAR	Security Assurance Requirement
SAS	Secure Attention Sequence
SD	Security Descriptor
SHA	Secure Hash Algorithm
SID	Security Identifier
SF	Security Functions
SFP	Security Functional Policy
SFR	Security Functional Requirement
SMB	Server Message Block
SP	Service Pack
SPI	Security Parameters Index
SRM	Security Reference Monitor
SSL	Secure Sockets Layer
ST	Security Target
SYSVOL	System Volume
TCP	Transmission Control Protocol
TDI	Transport Driver Interface
TLS	Transport Layer Security
TOE	Target of Evaluation
TPM	Trusted Platform Module
TSC	TOE Scope of Control
TSF	TOE Security Functions
TSS	TOE Summary Specification
UI	User Interface
UID	User Identifier
UNC	Universal Naming Convention
US	United States
URL	Uniform Resource Locator
USB	Universal Serial Bus
USN	Update Sequence Number
v5	Version 5
VDS	Virtual Disk Service
VPN	Virtual Private Network
VSS	Volume Shadow Copy Service
WAN	Wide Area Network
WebDAV	Web Document Authoring and Versioning
WU	Windows Update
WDM	Windows Driver Model
WMI	Windows Management Instrumentation
WSC	Windows Security Center

Abbreviation	Meaning
WWW	World-Wide Web
X64	A 64-bit instruction set architecture
X86	A 32-bit instruction set architecture

10 Appendix B—TOE Component Decomposition

Component	Subcomponent / Module
Administrator Tools Component	
	Active Directory Certificate Services Tools
	Active Directory Delegation of Control Wizard
	Active Directory Domains and Trusts Snap-in
	Active Directory Sites and Services Snap-in
	Audit Policy Command Line Interface
	Authorization Manager
	BitLocker Drive Encryption Control Panel
	Certificates Snap-in
	Component Services Snap-in
	Computer Management Snap-in
	Control Panel
	Create A Shared Folder Wizard
	Date and Time Control Panel
	Default Group Policy Object Restore Command Line Utility
	Device Manager Snap-in
	Devices and Printers Control Panel
	DHCP Snap-in
	Disk Management Snap-in
	DNS Snap-in
	Driver Verifier Manager
	Encrypting File System Dialog Boxes
	Event Viewer Snap-in
	Explorer
	Explorer Quota Property Tab
	File Encryption Command Line Utility
	Group Policy Editor Snap-in
	Group Policy Update Command Line Utility
	Hyper-V Manager
	Internet Information Service (IIS) Manager Snap-in
	IP Security Monitor Snap-in
	IP Security Policies Snap-in
	NAP Client Configuration Snap-in
	Network and Sharing Center Control Panel
	Performance Monitoring Snap-in
	Registry Editor
	Resultant Set of Policy Snap-in
	Routing and Remote Access Snap-in

Component	Subcomponent / Module
	SAM Lock Tool
	Schedule Service Command Line Interface
	Scheduled Tasks Command Line Utility
	Security Configuration Wizard
	Security Configuration Wizard Command Line Utility
	Security Policy Snap-in
	Security Templates Snap-in
	Security Configuration and Analysis Snap-in
	Server Manager
	Services Snap-in
	Shared Folders Snap-in
	Signature Verification Command Line Utility
	System Control Panel, Computer Name Tab
	System Integrity Check and Repair Command Line Utility
	Task Scheduler Snap-in
	TPM Management
	User Account Control Settings
	Users and Groups Snap-in
	Volume Shadow Copy Service Command Line Utility
	Windows Authentication User Interface
	Windows Firewall with Advanced Local Security Snap-in
	WMI Control Snap-in
Certificate Services Component	
	Certificate Service
	Certificate Service Default Exit Module
	Certificate Service Default Policy Module
	Online Responder Service
Cryptographic Support	
	BitLocker Drive Encryption Service
	FVE Crash Dump Driver
	FVE Driver
	TPM Base Services
	TPM Base Services DLL
	TPM Driver
Executive Component	
	64 bit Kernel Debug Support
	Advanced Local Procedure Call
	Application Compatibility Support
	Cache Manager
	Configuration Manager
	Event Tracing for Windows
	Executive Object Services
	Graphics Device Interface

Component	Subcomponent / Module
	Hardware Abstraction Layer
	Kernel Debug Manager
	Kernel Mode Windows Management Instrumentation
	Kernel Runtime
	Kernel Transaction Manager
	Memory Manager
	Microkernel
	Object Manager
	Plug and Play Manager
	Power Manager
	Process Manager
	Raw File System Library
	Security Reference Monitor
	Virtual DOS Machine
	Window Manager (User)
Hardware Component	
	AMD Hardware
	Intel Hardware
	Intel Itanium Hardware
Internet Information Server Component	
	BITS Server Extensions ISAPI
	IIS CoAdmin
	IIS ISAPI Handler
	IIS Metadata DLL
	IIS Reset Control
	IIS RPC Proxy
	IIS Web Admin Service
	IIS Web Server Core
	IIS Worker Process
	Internet Information Services
	ISAPI DLL for Web Printing
	Metadata and Admin Service
	Web Application Manager Registration
	WinHTTP Web Proxy Auto Discovery Service
IO: Core Component	
	CNG Kernel Cryptography
	File System Recognizer
	Generic Pass Through Driver
	I/O Manager
	Kernel Mode Driver Framework
	Kernel Mode Driver Framework Loader
	Kernel Security Device Driver
	Kernel Security Support Provider Interface Packages

Component	Subcomponent / Module
	Mount Manager
	User-mode Driver Framework Reflector
IO: Devices Component	
	ACPI Battery Miniclass Driver
	ACPI Driver
	ACPI Power Metering Driver
	Advanced Host Controller Interface Driver
	AMD Processor Driver
	Ataport Driver Extension
	Audio Port Class Driver
	Beep Driver
	Broadcom BCM5708C NetXtreme II GigE NIC Miniport Driver
	Broadcom NetXtreme 57xx Gb NIC Miniport Driver
	Composite Battery Driver
	File System Filter Manager
	Hardware Error Device Driver
	HID Class Library
	HID Keyboard Filter Driver
	HID Mouse Filter Driver
	HID Parsing Library
	HP ProLiant Smart Array
	i8042 Port Driver
	IDE ATAPI Port Driver
	IDE Mini-Port Drivers
	Intel Pro 1000 E1G60xx MT NIC Miniport Driver
	Intelligent IO Miniport Driver
	Intelligent IO Utility Filter Driver
	Intelligent Platform Management Interface Driver
	ISA and EISA Class Driver
	Keyboard Class Driver
	LSI Serial Attached SCSI Driver
	Microsoft System Management BIOS Driver
	Monitor Class Function Driver
	Mouse Class Driver
	Multipath Support Bus Drivers
	Null Driver
	NVIDIA nForce NIC Miniport Driver
	Parallel Port Driver
	Partition Manager
	Plug and Play PCI Enumerator
	Plug and Play Software Device Enumerator Driver
	PnP Disk Driver

Component	Subcomponent / Module
	PnP ISA Bus Driver
	Processor Device Driver
	SCSI CD-ROM Driver
	SCSI Class System Driver
	SCSI Port Driver
	SCSI Tape Class Driver
	SecureDigital Bus Driver
	Serial Device Driver
	Serial Port Enumerator
	Smart Card Driver Library
	Smart Card Reader Filter Driver
	Storage Port Driver
	USB 1.1 and 2.0 Port Driver
	USB CCID Driver
	USB Common Class Generic Parent Driver
	USB Host Controller
	USB Host Controller Interface Miniport Drivers
	USB Mass Storage Driver
	USB Miniport Driver for Input Devices
	USB Root Hub Driver
	User-Mode Bus Enumerator
	VDM Parallel Driver
	VGA Super VGA Video Driver
	Video Port Driver
	Volume Shadow Copy Driver
	Watchdog Driver
	Windows Management Interface for ACPI
IO: File Component	
	CD-ROM File System
	Encrypting File System
	Fast Fat File System
	File Information File System MiniFilter
	Mailslot Driver
	NPFS Driver
	NT File System Driver
	UDF File System Driver
	Volume Manager Driver and Extension Driver
IO: Net Component	
	Ancillary Function Driver for WinSock
	Client Side Caching Driver
	Computer Browser Datagram Receiver
	Distributed File System Client
	Distributed File System Filter Driver

Component	Subcomponent / Module
	FWP IPsec Kernel-Mode API
	HTTP Driver
	IP Filter Driver
	IP in IP Encapsulation Driver
	Kernel RPC Provider
	Loopback Network Driver
	Microsoft Tunnel Interface Driver
	Multiple UNC Provider
	NDIS User Mode IO Driver
	NDIS Wrapper Driver
	NetBT Transport Driver
	Network Store Interface Proxy Driver
	QoS Packet Scheduler Driver
	Redirected Drive Buffering Subsystem Driver
	Remote NDIS Miniport
	Server Network Driver
	SMB 1.0 Server Driver
	SMB 1.0 Sub-Redirector
	SMB 2.0 Server Driver
	SMB 2.0 Sub-Redirector
	SMB Mini-Redirector
	SMB Transport Driver
	TCPIP Protocol Driver
	TDI Translation Driver (TDX) Driver
	TDI Wrapper
	WebDav Mini Redirector
	Winsock 2 IFS Layer Driver
Network Support Component	
	COM Configuration Catalog Server
	COM Event System Service
	COM Services
	DHCP Service
	Distributed COM Services
	Domain Name Service
	Health Key and Certificate Management Service
	Internet Extensions for Win32
	Internet Key Exchange Service
	IP Helper Service
	IPSec SPD Server
	Network Access Protection Agent
	Network Connections Manager
	Network Location Awareness
	Network Policy Server

Component	Subcomponent / Module
	Network Store Interface Service
	NPS Host Support
	Quarantine Agent Proxy and Service Runtime
	Quarantine Client WMI Provider
	RPC Endpoint Mapper
	RPC Locator
	Simple TCP/IP Services Service DLL
	TCP/IP NetBIOS Transport Service
	TCP/IP Services Application
	Web DAV Service DLL
OS Support Component	
	Background Intelligent Transfer Service
	Distributed File System Service
	Print Spooler
	Removable Storage Manager
	Session Manager
	WMI Performance Reverse Adapter Service
	WMI Provider Host
	WMI Service
Security Component	
	Active Directory Replication Management
	Core Directory Service
	Credential Manager
	Credential Security Support Provider
	Data Protection API
	Directory Services Role Management
	Encrypting File System Service
	Inter-Site Messaging
	KDC Service
	Kerberos Security Package
	Key Isolation Service
	LDAP
	LSA Audit
	LSA Authentication
	LSA Policy
	MAPI Based Directory Request
	Microsoft Authentication Package V1.0
	Microsoft Base Smart Card Crypto Provider w/Infineon SICRYPT Card Module
	Microsoft Digest Access
	Microsoft Smart Card Key Storage Provider
	Microsoft Smart Card Minidriver
	Net Logon Service DLL

Component	Subcomponent / Module
	NT Directory Service Backup & Restore
	PKI Trust Installation and Setup
	Protected Storage Server
	SAM Server
	Secondary Logon Service
	TLS-SSL Security Provider
	Trust Signing APIs
	Windows Cryptographic Primitives Library
Services Component	
	Application Experience Lookup Service
	Application Information Service
	Certificate Propagation Service
	Computer Browser Service
	Cryptographic Services
	Desktop Windows Manager
	Diagnostic Policy Service
	File Replication Service
	Generic Host Process for Win32 Services
	Interactive Service Detection
	Non-COM WMI Event Provision APIs
	Offline Files Service
	Power Management Service
	Program Compatibility Assistant
	Remote Registry Service
	Server Service DLL
	Services and Controller App
	Smart Card Resource Management Server
	SuperFetch Service Host
	System Event Notification Service
	Task Scheduler Engine
	Universal Plug-and-Play Device Host
	User Mode Driver Framework Service
	User Profile Service
	User-mode Plug-and-Play Service
	Virtual Disk Service
	Volume Shadow Copy Service
	Windows Eventlog Service
	Windows Installer Service
	Windows Search
	Windows Security Center Service
	Windows Security Configuration Editor Engine
	Windows Shell Services DLL
	Windows Time Service

Component	Subcomponent / Module
	Windows Update Client
	Workstation Service
Virtualization	
	Hyper-V Infrastructure Driver
	Hyper-V Infrastructure Driver Library
	Hyper-V Image Management Service
	Hyper-V Networking Management Service
	Hyper-V Virtual Machine Management
	Hyper-V VMBus HID Miniport
	VHD Miniport Driver
	Virtual Machine Bus
Win32 Component	
	Base Server
	Client Server Runtime Process
	Windows Server DLL
Windows Firewall Component	
	Application Layer Gateway Service
	Base Filtering Engine Service
	Home Networking Configuration Manager
	IP Network Address Translator
	MAC Bridge Driver
	Network Address Translation Helper
WinLogon Component	
	Auto Enrollment
	Group Policy
	Group Policy Object Processing
	Local Session Manager
	Secure Desktop with Credential User Interface
	Syskey
	Trust Verification APIs
	Trusted Installer
	User Environment
	Windows File Protection
	Windows Logon Application
	Windows Logon User Interface Host
	Windows OS Startup – WinInit
	Windows OS Startup – WinLoad
	Windows OS Startup – WinResume
	Windows Smart Card Credential Provider