

Microsoft Windows 2003/XP Security Target

**Version 1.0
28 September 2005**

Prepared For:

Microsoft®
Microsoft Corporation
Corporate Headquarters
One Microsoft Way
Redmond, WA 98052-6399

Prepared By:

Science Applications International Corporation
Common Criteria Testing Laboratory
7125 Gateway Drive
Columbia, MD 21046-2554

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.

© 2005 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

1.	SECURITY TARGET INTRODUCTION.....	1
1.1	SECURITY TARGET, TOE, AND CC IDENTIFICATION.....	1
1.2	CC CONFORMANCE CLAIMS	2
1.3	STRENGTH OF ENVIRONMENT	2
1.4	CONVENTIONS, TERMINOLOGY, ACRONYMS	2
1.4.1	<i>Conventions</i>	3
1.4.2	<i>Terminology</i>	3
1.4.3	<i>Acronyms</i>	3
1.5	SECURITY TARGET OVERVIEW AND ORGANIZATION	3
2.	TOE DESCRIPTION.....	5
2.1	PRODUCT TYPES	5
2.2	PRODUCT DESCRIPTION	6
2.3	PRODUCT FEATURES	7
2.3.1	<i>Windows 2003/XP Administration and Management Features</i>	7
2.3.2	<i>Windows 2003/XP Network Security Features</i>	9
2.3.3	<i>Windows 2003/XP Scalability Features</i>	10
2.3.4	<i>New Windows 2003/XP TOE Features</i>	10
2.4	SECURITY ENVIRONMENT AND TOE BOUNDARY	16
2.4.1	<i>Logical Boundaries</i>	16
2.4.2	<i>Physical Boundaries</i>	18
2.5	TOE SECURITY SERVICES.....	18
3.	SECURITY ENVIRONMENT.....	20
3.1	THREATS TO SECURITY	20
3.2	ORGANIZATIONAL SECURITY POLICIES.....	21
3.3	SECURE USAGE ASSUMPTIONS.....	21
3.3.1	<i>Connectivity Assumptions</i>	21
3.3.2	<i>Personnel Assumptions</i>	22
3.3.3	<i>Physical Assumptions</i>	22
4.	SECURITY OBJECTIVES	23
4.1	TOE IT SECURITY OBJECTIVES	23
4.2	NON-IT SECURITY OBJECTIVES FOR THE ENVIRONMENT.....	24
5.	IT SECURITY REQUIREMENTS.....	25
5.1	TOE SECURITY FUNCTIONAL REQUIREMENTS.....	25
5.1.1	<i>Audit (FAU) Requirements</i>	31
5.1.2	<i>Cryptographic Support (FCS)</i>	35
5.1.3	<i>User Data Protection (FDP) Requirements</i>	39
5.1.4	<i>Identification and Authentication (FIA)</i>	47
5.1.5	<i>Management Requirements (FMT)</i>	49
5.1.6	<i>Protection of the TOE Security Functions (FPT)</i>	55
5.1.7	<i>Resource Utilization (FRU)</i>	57
5.1.8	<i>TOE Access (FTA)</i>	57
5.1.9	<i>Trusted Path/Channels</i>	58
5.2	TOE SECURITY ASSURANCE REQUIREMENTS.....	58
5.2.1	<i>Configuration Management (ACM)</i>	59
5.2.2	<i>Delivery and Operation (ADO)</i>	62
5.2.3	<i>Development (ADV)</i>	63
5.2.4	<i>Guidance Documents (AGD)</i>	67
5.2.5	<i>Life Cycle Support (ALC)</i>	69

5.2.6	<i>Security Testing (ATE)</i>	71
5.2.7	<i>Vulnerability Assessment (AVA)</i>	73
5.3	SECURITY REQUIREMENTS FOR THE IT ENVIRONMENT.....	76
6.	TOE SUMMARY SPECIFICATION	77
6.1	TOE SECURITY FUNCTIONS.....	77
6.1.1	<i>Audit Function</i>	77
6.1.2	<i>User Data Protection Function</i>	81
6.1.3	<i>Cryptographic Protection</i>	91
6.1.4	<i>Identification and Authentication Function</i>	94
6.1.5	<i>Security Management Function</i>	100
6.1.6	<i>TSF Protection Function</i>	104
6.1.7	<i>Resource Utilization Function</i>	111
6.1.8	<i>Session Locking Function</i>	111
6.2	TOE SECURITY ASSURANCE MEASURES.....	112
6.2.1	<i>Process Assurance</i>	112
6.2.2	<i>Delivery and Guidance</i>	113
6.2.3	<i>Design Documentation</i>	113
6.2.4	<i>Tests</i>	115
6.2.5	<i>Vulnerability Assessment</i>	116
7.	PROTECTION PROFILE CLAIMS	117
7.1	CONTROLLED ACCESS PROTECTION PROFILE (CAPP) CONFORMANCE CLAIM REFERENCE.....	117
7.1.1	<i>CAPP Requirements in ST</i>	117
7.1.2	<i>CAPP Differences and Enhancements</i>	117
8.	RATIONALE	123
8.1	SECURITY OBJECTIVES RATIONALE.....	123
8.1.1	<i>TOE IT Security Objectives Rationale</i>	123
8.1.2	<i>Non-IT Security Objectives for the Environment Rationale</i>	126
8.2	SECURITY REQUIREMENTS RATIONALE.....	127
8.2.1	<i>Security Functional Requirements Rationale</i>	127
8.2.2	<i>Security Assurance Requirements Rationale</i>	134
8.2.3	<i>Requirement Dependency Rationale</i>	134
8.2.4	<i>Explicitly Stated Requirements Rationale</i>	140
8.2.5	<i>Internal Consistency and Mutually Supportive Rationale</i>	141
8.2.6	<i>Strength of Function Rationale</i>	142
8.3	TOE SUMMARY SPECIFICATION RATIONALE.....	142
9.	ADDITIONAL PROTECTION PROFILE REFERENCES	146
9.1	PROTECTION PROFILE FOR SINGLE-LEVEL OPERATING SYSTEMS (SLOSPP) REFERENCE.....	146
9.2	WEB SERVER PP REFERENCE.....	146
	APPENDIX A LIST OF ACRONYMS	A-1
	APPENDIX B TOE COMPONENT DECOMPOSITION	B-1

1. Security Target Introduction

This section presents the following information:

- 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 2003/XP Security Target

ST Version – Version 1.0, 9/28/05

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

- Microsoft Windows Server 2003, Standard Edition (32-bit); Service Pack (SP) 1
- Microsoft Windows Server 2003, Enterprise Edition (32-bit and 64-bit versions); SP 1
- Microsoft Windows Server 2003, Datacenter Edition (32-bit and 64-bit versions); SP 1
- Microsoft Windows XP, Professional; SP 2
- Microsoft Windows XP, Embedded; SP 2

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

- MS05-042 – Vulnerabilities in Kerberos Could Allow Denial of Service (DoS), Information Disclosure, and Spoofing (899587)
- MS05-039 – Vulnerability in Plug and Play Could Allow Remote Code Execution and Elevation of Privilege (899588)
- MS05-027 – Vulnerability in Server Message Block (SMB) Could Allow Remote Code Execution (896422)
- A patch that updates the Internet Protocol (IP) Security (IPSec) Policy Agent is available for Windows Server 2003 and Windows XP (907865)

The following security updates must be applied to the above XP products:

- MS05-043 – Vulnerability in Print Spooler Service Could Allow Remote Code Execution (896423)
- MS05-042 – Vulnerabilities in Kerberos Could Allow DoS, Information Disclosure, and Spoofing (899587)
- MS05-039 – Vulnerability in Plug and Play Could Allow Remote Code Execution and Elevation of Privilege (899588)
- MS05-027 – Vulnerability in SMB Could Allow Remote Code Execution (896422)
- MS05-018 – Vulnerability in Windows Kernel Could Allow Elevation of Privilege and DoS (890859)
- MS05-012 – Vulnerability in Object Linking and Embedding (OLE) and Component Object Model (COM) Could Allow Remote Code Execution (873333)
- MS05-011 – Vulnerability in SMB Could Allow Remote Code Execution (885250)

- MS05-007 – Vulnerability in Windows Could Allow Information Disclosure (888302)
- MS04-044 – Vulnerabilities in Windows Kernel and Local Security Authority Subsystem Service (LSASS) Could Allow Elevation of Privilege (885835)
- A hotfix that updates the IPSec Policy Agent is available for Windows Server 2003 and Windows XP (907865)

TOE Hardware Identification – The following hardware platforms are included in the evaluated configuration:

- HP ProLiant DL380 G3 X2.8 Gigahertz (GHz)
- HP rx2600 1.5GHz Central Processing Unit (CPU) Server Solution
- HP Workstation ZX2000
- Dell Optiplex GX270
- Unisys ES700-420 (64-bit)
- Unisys ES7000-540-G3 (32-bit)
- Infineon SICRYPT Smart Cards
- IBM xSeries 346

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

CC Identification – CC for Information Technology (IT) Security Evaluation, Version 2.2, January 2004.

International Standard – International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 15408:1999.

Keywords – OS, sensitive data protection device, directory service, network management, desktop management, single sign on, Discretionary Access Control (DAC), ST, embedded, cryptography, Public key, firewall, web server, IPSec, smart card, certificate server, IP Version 6 (IPv6), information flow, Federal Information Processing Standard (FIPS)-140, Virtual Private Network (VPN), content-provider, access control, Controlled Access Protection Profile (CAPP), EAL 4, Microsoft Windows, 32 bit, 64 bit.

1.2 CC Conformance Claims

This TOE and ST are consistent with the following specifications:

- Conformant to PP, Controlled Access Protection Profile, Version 1.d, National Security Agency, 8 October 1999 (PP Conformant). Note that the CAPP requires EAL3.
- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 2.2, January 2004, extended (Part 2 extended)
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.2, January 2004, conformant, EAL4 augmented with ALC_FLR.3 (Part 3 Conformant, EAL 4 augmented).

1.3 Strength of Environment

The evaluation of Windows 2003/XP provides a moderate level of independently assured security in a conventional TOE and is suitable for the environment specification in this ST. The assurance requirements and the minimum Strength of Function (SOF) were chosen to be consistent with this goal and to be compliant with the CAPP. The TOE assurance level is EAL 4 augmented with ALC_FLR.3 and the TOE minimum SOF is SOF-medium.

1.4 Conventions, Terminology, Acronyms

This section specifies the formatting information used in the ST.

1.4.1 Conventions

The following conventions have been applied in this document:

- 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. In the ST, a letter placed at the end of the component indicates iteration. For example FMT_MTD.1 (a) and FMT_MTD.1 (b) indicate that the ST includes two iterations of the FMT_MTD.1 requirement, a and b.
 - 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 bolding to highlight text of special interest, such as captions.

1.4.2 Terminology

The following terminology is used in the ST:

- Authorized User – an entity that has been properly identified and authenticated. These users are considered to be legitimate users of the TOE.
- Authorized administrator/Administrator – A user in the administrator role is an authorized user who has been granted the authority to manage the TOE. These users are expected to use this authority only in the manner prescribed by the guidance given them. The term authorized administrator is taken from the CC and CAPP and is used in the ST in those sections that are derived from the CAPP or the CC directly. Otherwise, the term administrator is used. These terms are used interchangeably.
- DAC Policy – The DAC policy is defined as in the CAPP.

1.4.3 Acronyms

The acronyms used in this ST are specified in Appendix A – Acronym List.

1.5 ST Overview and Organization

The Windows 2003/XP TOE is a general-purpose, distributed, network OS that provides controlled access between subjects and user data objects. Windows 2003/XP 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, Transport Layer Security (TLS)/Secure Sockets Layer (SSL), Digest, Internet Key Exchange (IKE)/IPSec, FIPS-140 validated cryptography, web service, and Light-weight Directory Access Protocol (LDAP) Directory-based resource management. The Windows 2003/XP TOE provides the following security services: user data protection (WEBUSER access control, web content provider access control, DAC, IPSec information flow control, connection firewall information flow control), cryptographic support, audit, Identification and Authentication (I&A) (including trusted path/channel), security management, protection of the TOE Security Functions (TSF), resource quotas, and TOE access/session. The Windows 2003/XP 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 incoming and out-going traffic channels through a physically separate part of the TOE. The TOE is capable of auditing security relevant events that occur within a Windows 2003/XP network. All these security controls require users to identify themselves and be authenticated prior to using any node on the network.

The Windows 2003/XP ST contains the following additional sections:

- TOE Description (Section 2) – Provides an overview of the TSF and boundary.
- Security Environment (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 environment.
- IT 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 CAPP.
- Rationale (Section 8) – Presents the rationale for the security objectives, requirements, and TOE Summary Specifications (TSS) as to their consistency, completeness and suitability.
- Additional PP References (Section 9) – Summarizes content drawn for other unclaimed PPs above and beyond that drawn from the CAPP.

2. TOE Description

The TOE includes the Windows 2003/XP OS, supporting hardware, and those applications necessary to manage, support and configure the OS.

2.1 Product Types

Windows 2003/XP is a preemptive multitasking, multiprocessor, and multi-user OS. In general, OS' 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 2003/XP expands these basic OS capabilities to controlling the allocation and managing higher level IT resources such as security principals like user or machine accounts, files, printing objects, services, windowstation, desktops, cryptographic keys, network ports/traffics, directory objects, and web contents. Multi-user OS' such as Windows 2003/XP, 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 2003/XP provides an interactive User Interface (UI), as well as a network interface. The TOE includes a homogenous set of Windows 2003/XP systems that can be connected via their network interfaces and may be organized into domains. A domain is a logical collection of Windows 2003/XP systems that allows the administration and application of a common security policy and the use of a common accounts database. Windows 2003/XP supports single and multiple domain configurations. In a multi-domain configuration, the TOE supports implicit and explicit trust relationships between domains. Domains use established trust relationships to share account information and validate the rights and permissions of users. A user with one account in one domain can be granted access to resources on any server or workstation on the network. Domains can have one-way or two-way trust relationships. 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 2003/XP 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 2003/XP 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 ST, 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.

Other than an OS Windows 2003/XP can also be categorized as the following types of **Information Assurance (IA)** or IA enabled IT products:

- Windows 2003/XP is a **Sensitive Data Protection Device** to defend the Computing Environment. The core mechanism in this case is the Windows 2003/XP Encrypting File System (EFS), which is part of the Windows 2003/XP TOE.
- Windows 2003/XP is a **Directory Service** product to support Security Infrastructure. The LDAP based access and management of Windows Active Directory (AD) objects is part of the Windows 2003/XP T Interfaces (TSFI).
- Windows 2003/XP is a **Network Management** product to support the Security Infrastructure. Windows 2003/XP Group Policy, which is part of the Windows 2003/XP TOE and provides the network management in Windows 2003/XP networks.

- Windows 2003/XP is a **Desktop Management** product to support the Security Infrastructure. Windows 2003/XP Group Policy Service, which is part of Windows 2003/XP TOE and provides the desktop management of Windows 2003/XP TOE desktops.
- Windows 2003/XP is a **Single Sign On** product (using password or smart card) for Windows 2003/XP networks to defend the Computing Environment. Windows 2003/XP supports single sign on to the TOE.
- Windows 2003/XP 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 2003/XP is a **VPN** product providing an IPSec service and its associated Transport Driver Interface (TDI) based network support.
- Windows 2003/XP is an **Embedded OS** product in that it that delivers the power of the Windows OS in componentized form, enabling embedded developers to build reliable and advanced embedded devices.
- Windows 2003 is a **Web Server** product by including the Internet Information Services (IIS) Version 6.0 (IIS6) component functionality which provides a web service application infrastructure utilizing the underlying OS services.

2.2 Product Description

Windows 2003/XP is an OS that supports both workstation and server installations. The TOE includes five product variants of Windows 2003/XP: XP Embedded, XP Professional, Server 2003 Server, Server 2003 Enterprise Server, and Server 2003 Data Center. The server products additionally provide DC features including the AD and Kerberos Key Distribution Center (KDC). The server products in the TOE also provide IIS6, Certificate Server, File Replication, Directory Replication, Domain Name System (DNS), Dynamic Host Configuration Protocol (DHCP), Distributed File System (DFS) service, Removable Storage Manager, and Virtual Disk Service. The XP Embedded product in the TOE provides the Enhanced Write Filter (EWF) feature to allow the OS to boot from a disk volume residing on any read-only media such as flash media, Compact Disk—Read Only Memory (CD-ROM), or a write-protected hard drive. Otherwise, all variants include the same security features. The primary difference between the variants is the number of users and types of services they are intended to support.

Windows 2003/XP Professional are suited for business desktops and notebook computers (note that only desktops are included in the evaluated configuration); it is the workstation product. Designed for departmental and standard workloads, Windows Server 2003 Standard Server delivers intelligent file and printer sharing; secure connectivity based on Internet technologies, and centralized desktop policy management. Windows Server 2003 Enterprise Server differs from Windows Server 2003 Standard Server primarily in its support for high-performance servers for greater load handling. These capabilities provide reliability that helps ensure systems remain available. Windows Server 2003 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 XP Embedded is the embedded OS that delivers the power of Windows in componentized form for rapidly building reliable and advanced embedded devices.

The security features addressed by this security target are those provided by Windows 2003/XP as an OS. Microsoft provides several Windows 2003/XP 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, Terminal Service, Microsoft Message Queue, Microsoft Distributed Transaction Coordinator, Right Management Service, and Windows SharePoint Service. The features identified and described in this section are included in the TOE and as such are within the scope of the evaluation.

The following table summarizes the TOE configurations included in the evaluation. There are eleven stand-alone configurations and seventeen networked configurations.

	Windows XP Professional (32 bit)	Windows XP Embedded (32 bit)	Windows Server 2003 Standard (32 bit)	Windows Server 2003 Enterprise (32 bit)	Windows Server 2003 Enterprise (64 bit)	Windows Server 2003 Data Center (32 bit)	Windows Server 2003 Data Center (64 bit)
Single Processor	X	X	X	X	X	N/A	N/A
Multiple Processor	X	N/A	X	X	X	X	X
Stand-alone	X	X	X	X	X	X	X
Domain Member	X	X	X	X	X	X	X
Domain Controller	N/A	N/A	N/A	X	X	X	X
Variations as a Domain Element	2	1	2	4	4	2	2

2.3 Product Features

Windows 2003/XP has many features, several of which support simplifying the administration and management of a distributed environment, in order to improve network security, and scalability. This section highlights several of these features.

2.3.1 Windows 2003/XP Administration and Management Features

Windows 2003/XP distributed security features provide scalable, flexible account management for large domains with fine-grain access control and delegation of administration. A few of these administration features are briefly described below.

Group Policy

Windows 2003/XP 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.

Upon installation, Windows 2003/XP offers groups that are pre-configured with specific user rights and/or privileges. These groups are referred to as "built-in groups." The Windows 2003/XP 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); 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.

Delegated Administration

Windows 2003/XP introduces 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. Delegation helps companies reduce the number of domains they need to support a large organization with multiple geographical locations.

AD can interoperate or synchronize data with other directory services using LDAP, Microsoft Directory Service (DS) Synchronization, or AD Connector.

Access Control Lists (ACLs)

Windows 2003/XP permits 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 ACL. An ACL is a list of entries that grant or deny specific access rights to individuals or groups. The Windows 2003/XP Server object-based security model lets administrators grant access rights to a user or group-rights that govern who can access a specific object, a group of properties, or an individual property of an object. The definition of access rights on a per-property level provides the highest level of granularity of permissions.

Disk Quotas

Windows 2003/XP allows 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.

Windows Management Instrumentation (WMI)

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.

Administrative Tools

Windows 2003/XP delivers an integrated set of management tools and services. Only a few are described below.

Setup Manager: provides a graphical wizard that guides authorized administrators in designing installation scripts.

Backup and Recovery: Windows 2003/XP backup and recovery features make it easier to backup data and then recover data in the event of a hard disk failure. Windows 2003/XP allows back up to a single file on a hard disk or to external media.

Administrative wizards: Windows 2003/XP makes it easier to perform routine or challenging tasks, resulting in fewer help desk calls and better customer service. For example, the Add Printer Wizard makes it easy to connect to local and network printers even when you're browsing the network.

Microsoft Management Console (MMC)

MMC unifies and simplifies system management tasks through a central, customizable console that allows control, monitoring, and administration of widespread network resources. All management functions in Windows 2003/XP are available through the MMC snap-ins (including AD Domains and Trusts, AD Sites and Services, AD Users and Computers, Component Services, Computer Management, Certificate

Management, Event Viewer, Group Policy, IPSec Policy Management, Security Template, Security Configuration and Analysis).

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 2003/XP identifies device drivers that have passed the Windows Hardware Quality Labs test and warns users if they are about to install an uncertified driver.

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 of displaying information and receiving user input. Graphical applications use resources, such as windows to display information and receive 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: a Window station object and a Desktop Object. Each is an object with a DACL that is used to control access to it.

2.3.2 Windows 2003/XP Network Security Features

Windows 2003/XP Server secures network data using an authentication protocol. For an additional level of security within a site, network data can also be encrypted. All network communication can be encrypted for specific clients, or for all clients in a domain using IPSec. Several features that support improved network security are briefly described below.

EFS

Windows 2003/XP increases security of data on the hard disk by encrypting it. This data remains encrypted even when backed up or archived. 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.

Kerberos Authentication Support

Full support for Kerberos Version 5 (v5) protocol Windows 2003/XP provides fast, single sign-on to Windows 2003/XP-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.

Support for Security Standards

Windows 2003/XP builds secure network sites using the latest standards, including 128-bit SSL/TLS, IPSec and Kerberos v5 authentication.

Secure Network Communications

Windows 2003/XP supports 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.

Cryptographic API (CryptoAPI)

Windows 2003/XP CryptoAPI provides applications access to FIPS-140 compliant cryptographic functions, public keys, credential management and certificate validation functions.

Dynamic DNS

The AD integrated, Internet standards-based DNS service simplifies object naming and location through Internet protocols, and improves scalability, performance and interoperability. Systems that receive addresses from a DHCP server are automatically registered in DNS. Replication options through AD can simplify and strengthen name replication infrastructure.

2.3.3 Windows 2003/XP Scalability Features

Windows 2003/XP delivers scalability features that support higher volumes of users and more complex applications. Several of these features are described below.

Memory and Processor Support

Windows 2003/XP Professional and Windows 2003/XP Server support up to four (4) Gigabytes (GBs) of Random Access Memory (RAM) and up to four (4) symmetric multiprocessors. Windows Server 2003 Enterprise Server takes advantage of larger amounts of memory to improve performance and handle the most demanding applications, with support for up to 32 GB of RAM for x86-based computers and 64 GB of RAM for Itanium-based computers. It supports up to eight (8) symmetric multiprocessors. Windows Server 2003 Data Center supports 64 GB of RAM for x86-based computers and 512 GB of RAM for Itanium-based computers. It takes care of a maximum 64 symmetric multiprocessors.

High Throughput and Bandwidth Utilization

Windows Server 2003 includes 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 2003 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.

Job Object API

The Windows 2003/XP Job Object API, with its ability to setup processor affinity, establish time limits, control process priorities, and limit memory utilization for a group of related processes, allows 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.

DFS

Windows 2003/XP 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.

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 network is a peer of all other replicas; changes can be made to any replica and will be reflected across all of them.

2.3.4 New Windows 2003/XP TOE Features

The below highlight additional features that were not available in previous Windows 2000 CC evaluation completed in Oct 2002, but are included in Windows 2003/XP TOE.

Volume Shadow Copy Service (VSS)

VSS coordinates shadow copies for applications and target New Technology File System (NTFS) volumes in a point-in-time copy. Through integration with target applications such as AD VSS enables consistent state management. If, during the shadow copy creation process, an application writes to disk, data inconsistencies can occur, compromising the integrity of the point-in-time data image. The VSS has been designed to eliminate these drawbacks.

Fault-Tolerant Process Model and Kernel-Mode Web Driver

With IIS6, 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. In addition to delivering significantly better Web server throughput, the new architecture of IIS6 significantly improves Web server stability because a single faulty application running on the Web server can no longer bring down other applications on the same server. The worker process that is servicing the faulty application can simply be recycled without affecting other worker processes.

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 raw encrypted file to the WebDAV server using an HyperText Transfer Protocol (HTTP) “PUT” command. Similarly, encrypted files downloaded to a client are transmitted as raw encrypted files using an HTTP “GET” command and decrypted locally on the client.

Forest Trust

Forest trust is a new type of Windows trust for managing the security relationship between two 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, without sacrificing the single sign-on benefits of having only one user Identification (ID) and password maintained in the user’s home forest.

IPv6

IPv6 is the next generation of the Internet layer protocols of the Transmission Control Protocol (TCP)/IP protocol suite. The IPv6 protocol driver includes utilities and API support for IPv6-enabled system components. Windows IPv6 also provides support for IPv6/IP Version 4 (IPv4) coexistence technologies such as “6to4” and Intra-site Automatic Tunnel Addressing Protocol (ISATAP).

Smart Card Support for Authentication Digest Authentication

Smart Card technology is fully integrated into the Windows XP/2003 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 XP/2003 TOE to authenticate users by using the private and public key information stored on a card. The Smart Card subsystem on the Windows XP/2003 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 RS-232 and Universal Serial Bus (USB). The Windows XP/2003 TOE detects Plug and Play-compliant smart card readers and installs them using the Add Hardware wizard.

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 Windows XP/2003 TOE. Digest authentication does not have to use reversible password encryption. With Digest authentication in Windows 2000, a user who is authenticated by using Digest authentication must have the password stored with the Store password using reversible encryption option flagged. However, in Windows Server 2003 TOE, 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.

Public Key Certificate Issuing and Management Service

The Windows Server 2003 Certificate Server issues and manages public key certificates for the following Windows XP/2003 TOE services: digital signatures, software code signing, TLS/SSL authentication for Web traffic, IPsec, Smart card logon, EFS user and recovery certificates.

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.

Virtual Disk Service (VDS)

VDS provides a set of utilities for managing the hardware disks. VDS implements a single, uniform interface for managing disks. Each hardware vendor writes a VDS provider that translates the general purpose VDS APIs into specific instructions for their hardware. Windows 2003/XP includes VDS providers for basic and dynamic disks.

Client Side Caching Off-line Files Support with SMB/Common Internet File System (CIFS) Redirector

When Windows 2003/XP client is caching a file and the Windows 2003/XP 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. If the Windows 2003/XP file server is not available, the client with the SMB/CIFS Redirector also has the cached copy to use.

Windows XP Embedded EWF

Windows XP Embedded has an EWF that allows for writes to be made to memory or a special EWF partition that an administrator can set. The administrator can also turn off the filter to make changes to the protected volume, either programmatically with a special EWF API or by using the EWF management utilities. The EWF allows the OS to boot from a disk volume residing on any read-only media such as flash media, CD-ROM, or a write-protected hard drive while maintaining the appearance of read/write access to the OS.

IIS6 Web Service

An IIS6 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 "HTTP.sys" for sending requests and receiving responses over HTTP. Depending on how IIS6 is configured, there can be multiple worker processes running, serving different Web applications concurrently. This design separates applications by process boundaries, and it helps achieve maximum Web server reliability and security.

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

Windows Firewall is a stateful firewall that drops unsolicited incoming traffic that does not correspond to either traffic sent in response to a request of the computer (solicited traffic) or 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 XP/2003 TOE 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 run-time ICF policy and remove the boot-time filters.

64-bit Support

Windows Server 2003 Enterprise Server and data Center include support for machines with new 64-bit CPUs. In addition to processing twice as much data per clock cycle and providing more on-chip computing

resources (such as registers) than their 32-bit counterparts, 64-bit CPUs eliminate many of the inherent resource management limitations of 32-bit architectures. By enabling more system resources to be added and to be used more efficiently, 64-bit servers deliver significantly greater levels of performance and scalability.

Storport Driver

Windows 2003/XP introduces a new port driver called Storport (storport.sys), which delivers significantly greater disk I/O processing efficiency and throughput, especially when used with high-performance devices such as host-based Redundant Array of Independent Disks (RAID) and fiber-channel adapters. There are several advantages to using the Storport driver, including reduced system resource usage and better performance. Some of the primary reasons for the Storport driver's better performance and resource usage include: Full-Duplex Mode, Reduced Device Lock Contention, Increased Queuing Efficiency.

Multiple DFS Roots

Windows Server 2003 Enterprise Server and Datacenter can support multiple DFS root directories on a single server (Windows 2000 is limited to a single DFS root per server).

Larger Directory Database Cache

AD implements an in-memory cache that resides in user space and stores directory objects for faster access than if they had to be retrieved from disk. In Windows 2000, this cache was limited to 512 Megabytes (MBs) under normal conditions and 1024 MB when the /3GB switch was used. In Windows Server 2003, this cache is allowed to grow more freely, although it is still limited by the amount of virtual address space (approximate maximum sizes are 2.2 GB with the /3GB switch and 1.5 GB without the switch). With the cache able to store more objects, cache hit ratios are higher and performance is improved.

Auto-enrollment

Public Key Certificate auto-enrollment and auto-renewal in Windows Server 2003 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.

Delta Certificate Revocation Lists (CRLs)

The certificate server included in Windows Server 2003 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.

Cross-Certification Support

Also called qualified subordination¹, 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.

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.

Constrained Delegation

Delegation is the act of allowing a service to impersonate a user account or computer account in order to access resources throughout the network. This new feature in Windows Server 2003 enables you to limit delegation to specific services, to control the particular network resources the service or computer can use.

¹ Qualified subordination is different from "qualified certificates" defined in RFC 3739.

For example, a service that was previously trusted for delegation in order to access a backend on behalf of a user can now be constrained to use its delegation privilege only to that backend and not to other machines or services.

Protocol Transition

In Windows Server 2003 TOE, the new 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. Thus a user can be authenticated using an alternative authentication method and then obtain a Windows identity, subject to system policy.

Improvements in IPSec

Windows Server 2003 TOE includes support for a stronger cryptographic master key, the use of a 2048-bit Diffie-Hellman (DH) key exchange. In Windows XP and the Windows Server 2003 family, IP Security Monitor is implemented as part of MMC and includes enhancements that allow you to monitor IPSec information for your local computer and for remote computers. In addition, it is now possible to create and assign a persistent IPSec policy to secure a computer if a local IPSec policy or an AD-based IPSec policy cannot be applied.

EFS Multi-user Support

Windows XP/2003 TOE supports 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.

File Replication Service (FRS)

FRS is a technology that replicates files and folders stored in the System Volume (SYSVOL) shared folder on domain controllers and 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.

Globally Unique Identifier (GUID) Partition Table (GPT)

The 64-bit versions of Windows Server 2003 Enterprise Server and Datacenter support a new disk partitioning style, 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.

Password Backup and Restore Service

A new Password Backup and Restore Service makes it easy for users to create a backup disk that can be used to reset their password. The service provides users with a secure mechanism for resetting their password without administrative intervention. The password is not stored on the backup disk. The disk can be used only to reset the password for the associated user account.

Network Address Translation (NAT)

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 letting you use 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 XP/2003 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 XP/2003 TOE can recognize and adapt to hardware configuration changes automatically, without user intervention and or restarting the computer.

Increased Performance for Network Printing

An enhanced standard port monitor in print spooler of the Windows XP/2003 TOE provides a fast and robust method for printing to network-attached printers and provides better performance and richer device status. Other enhancements include support for print drivers that can be downloaded automatically when client computers connect to print servers, a benefit that simplifies printing for users and administrators.

COM Plus Component Service Infrastructure

COM Plus Component Service is an Infrastructure running the Windows XP/2003 TOE based on extensions of the. COM Plus Component Service provides threading and security, object pooling, queued components, and application administration and packaging.

Internet Connection Sharing (ICS)

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 ILAN to the WAN.

Application Compatibility Support

Application Compatibility technology provides an environment for running programs that more closely reflects the behavior of previous Microsoft OS releases. Application compatibility technology consists of a user mode service and kernel mode cache support. The service defines an external interface to the application compatibility cache support. The cache resides in system space and is mapped into the address space of every process.

Kernel Debug Management

The Kernel Debugger subcomponent supports authorized users to debug running processes in the Windows XP/2003 TOE by allowing them to attach a debugger to a running process via a new 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.

“Winsock2” Installable File System (IFS) Layer Driver

The “Winsock2” 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.

Processor Run Time Power Management

For each family of processors supported by the Windows XP/2003 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.

Removable Storage Manager

Removable Storage Manager of the Windows XP/2003 TOE makes it easy to track removable storage media (tapes and optical discs) and to manage the hardware libraries, such as changers and jukeboxes, that contain them. Note that, currently, hardware changers and jukeboxes are not parts of the TOE.

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.

Windows Security Center Service (WSC)

WSC is a service that monitors the status of Windows firewall running on the Windows XP/2003 TOE. It also provides the logged-on interactive user certain visual notifications when it detects that the status of Windows firewall has changed.

Web Site Permissions

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 IIS6 web server of the Windows Server 2003 TOE. An authorized user can configure web site's access permissions for specific sites, directories, and files. 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 XP Embedded Bootable CD-ROM Disk Image Driver

Windows XP Embedded TOE includes the El Torito driver. This driver implements the El Torito specification, which provides a standard for creating a bootable CD-ROM as an image of a hard disk drive or a floppy disk drive. When an image of a hard disk is created, the CD-ROM boots as drive "C" and all hard disk drive letters are shifted up one letter.

Windows Installer Service

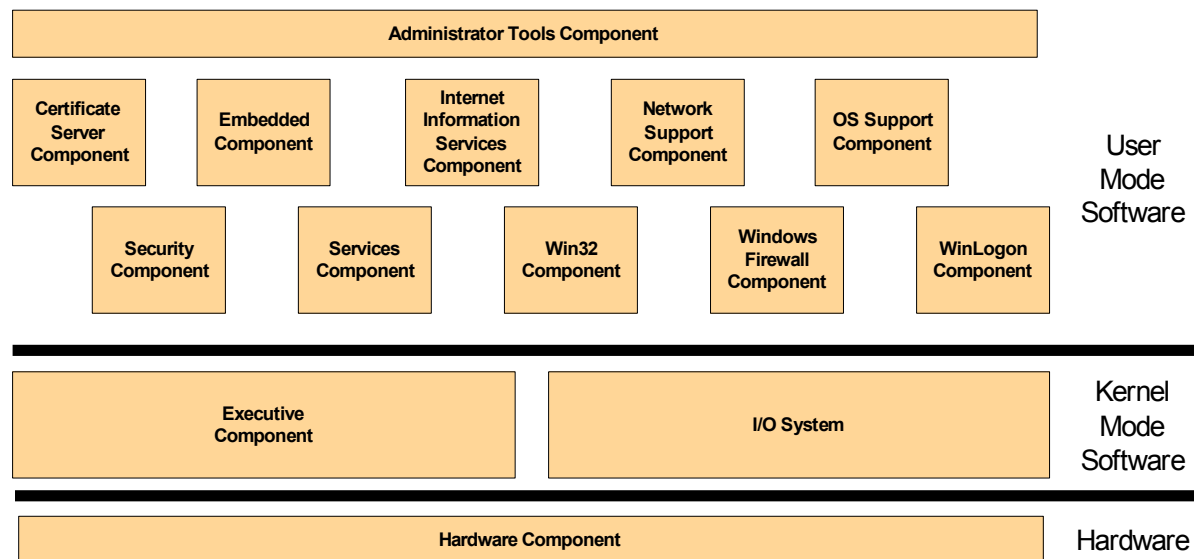
The Windows Installer Service enables customers to better address corporate deployment and provide a standard format for component management. The installer supports advertisement of applications and features according to the operating system settings. It can install multiple patches with a single transaction 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.

2.4 Security Environment and TOE Boundary

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

2.4.1 Logical Boundaries

The diagram below depicts components and subcomponents of Windows 2003/XP that comprise the TOE. The components/subcomponents are large portions of the Windows 2003/XP OS, and generally fall along process boundaries and a few major subdivisions of the kernel mode OS.



The system components are:

- Administrator Tools Module
 - Administrator Tools Component (aka GUI 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).
- Embedded Module
 - Embedded Component: The embedded component provides a variety of applications that facilitate the OS functioning in devices that require an embedded OS.
- 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

- Miscellaneous OS Support Module
 - OS Support Component: This component is a set of processes that provide various other OS support functions and services
- Remote Procedure Call (RPC) and 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.
- Web 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.

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 Intel X86 or IA64 machine or equivalent processor (including Pentium family) 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 Data Center product. A set of devices may be attached and they are listed as follows:

- Display Monitor,
- Keyboard,
- Mouse,
- Floppy Disk Drive,
- CD-ROM Drive
- Fixed Disk Drives,
- Printer,
- Audio Adaptor,
- Network Adaptor, and
- Smart Card Reader.

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

The security services provided by the TOE are summarized below:

- **Security Audit** – Windows 2003/XP has 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 date and time of the event, user who caused the event to be generated, computer where the event occurred, and other event specific data. Authorized administrators can review audit logs.
- **Identification and Authentication** – Windows 2003/XP requires 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 2003/XP maintains a database of accounts including their identities, authentication information, group associations, and privilege and logon rights associations. Windows 2003/XP includes a set of account policy functions that include the ability to define minimum password length, number of failed logon attempts, duration of lockout, and password age.
- **Security Management** – Windows 2003/XP includes 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 2003/XP protect user data by enforcing several access control policies (DAC, WEBUSER and web content provider access control) and several information flow policies (IPSec filter information flow control, Connection Firewall); and, object and subject residual information protection. Windows 2003/XP uses access control methods to allow or deny access to objects, such as files, directory entries, printers, and web content. Windows 2003/XP uses information flow control methods to control the flow of IP traffic and packets. It 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 2003/XP also protects user data by ensuring that resources exported to user-mode processes do not have any residual information.
- **Cryptographic Protection** - Windows 2003/XP provides additional protection of data through the use of data encryption mechanisms. These mechanisms only allow authorized users access to encrypted data.
- **Protection of TOE Security Functions** – Windows 2003/XP provides a number of features to ensure the protection of TOE security functions. Windows 2003/XP protects against unauthorized data disclosure and modification by using a suite of Internet standard protocols including IPSec and ISAKMP. Windows 2003/XP ensures process isolation security for all processes through private virtual address spaces, execution context and security context. The Windows 2003/XP data structures defining process address space, execution context, and security context are stored in protected kernel-mode memory.
- **Resource Utilization** – Windows 2003/XP 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 2003/XP provides the ability for a user to lock their session immediately or after a defined interval. It constantly monitors the mouse and keyboard for activity and locks the workstation after a set period of inactivity. Windows 2003/XP allows an authorized administrator to configure the system to display a logon banner before the logon dialogue.

3. Security Environment

The TOE security environment consists of the threats to security, organizational security policies, and usage assumptions as they relate to Windows 2003/XP. The assumptions and policies are primarily derived from the CAPP, while the threats have been introduced to better represent specific threats addressed by Windows 2003/XP.

3.1 Threats to Security

Table 3-1 presents known or presumed threats to protected resources that are addressed by Windows 2003/XP.

Table 3-1 Threats Addressed by Windows 2003/XP

Threat	Description
T.AUDIT_CORRUPT	Unauthorized users may tamper with audit data or unauthorized users may cause audit data to be lost due to failure of the system to protect the audit data.
T.CONFIG_CORRUPT	Configuration data or other trusted data may be tampered with by unauthorized users due to failure of the system to protect this data.
T.OBJECTS_NOT_CLEAN	Users may request access to resources and gain unauthorized access to information because the system may not adequately remove the data from objects between uses by different users, thereby releasing information to the subsequent user.
T.SPOOF	A hostile entity masquerading as the IT system may receive unauthorized access to authentication data from authorized users who incorrectly believe they are communicating with the IT system during attempts by a user to initially logon.
T.SYSACC	An unauthorized user may gain unauthorized access to the system and act as the administrator or other trusted personnel due to failure of the system to restrict access.
T.UNAUTH_ACCESS	An unauthorized user may gain access to system data due to failure of the system to restrict access.
T.UNAUTH_MODIFICATION	An unauthorized user may cause the modification of the security enforcing functions in the system, and thereby gain unauthorized access to system and user resources due to failure of the system to protect its security enforcing functions
T.UNDETECTED_ACTIONS	An unauthorized user may perform unauthorized actions that go undetected because of the failure of the system to record actions.
T.USER_CORRUPT	User data may be tampered with by unauthorized users due to failure of the system to enforce the restrictions to data specified by authorized users.
T.ADMIN_ERROR	An administrator may incorrectly install or configure the TOE resulting in ineffective security mechanisms.
T.AUDIT_COMPROMISE	A malicious process or user may cause audit data to be inappropriately accessed (viewed, modified or deleted), or prevent future records from being recorded, thus masking an attacker's actions.
T.EAVESDROP	A malicious process or user may intercept data transmitted within the enclave.
T.MASQUERADE	An unauthorized user, process, or external IT entity may masquerade as an authorized entity to gain access to data or TOE resources.
T.POOR_DESIGN	Unintentional or intentional errors in requirement specification, design or development of the TOE may occur.

Threat	Description
T.POOR_IMPLEMENTATION	Unintentional or intentional errors in implementing the design of the TOE may occur.
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.UNATTENDED_SESSION	A user may gain unauthorized access to an unattended session.
T.UNIDENTIFIED_ACTIONS	Failure of the administrator to identify and act upon unauthorized actions may occur.
T.ADDRESS_MASQUERADE	A user on one interface may masquerade as a user on another interface to circumvent the TOE policy.
T.TCPIP_ATTACK	A threat agent may take advantage of a published vulnerability against protocols layers below HTTP (e.g. TCP or IP), resulting in the TOE being unable to respond properly to valid requests.

3.2 Organizational Security Policies

Table 3-2 describes organizational security policies that are addressed by Windows 2003/XP.

Table 3-2 Organizational Security Policies

Security Policy	Description	PP Source
P.ACCOUNTABILITY	The users of the system shall be held accountable for their actions within the system.	CAPP
P.AUTHORIZED_USERS	Only those users who have been authorized access to information within the system may access the system.	CAPP
P.NEED_TO_KNOW	The system must limit the access to, modification of, and destruction of the information in protected resources to those authorized users which have a "need to know" for that information.	CAPP
P.AUTHORIZATION	The system must have the ability to limit the extent of each user's authorizations.	
P.ADD_IPSEC	The system must have the ability to protect system data in transmission between distributed parts of the protected system	
P.WARN	The system must have the ability to warn users regarding the unauthorized use of the system.	

3.3 Secure Usage Assumptions

This section describes the security aspects of the environment in which Windows 2003/XP is intended to be used. This includes assumptions about the connectivity, personnel, and physical aspects of the environment.

Windows 2003/XP is assured to provide effective security measures in the defined environment only if it is installed, managed, and used correctly. The operational environment must be managed in accordance with the user and administrator guidance.

3.3.1 Connectivity Assumptions

Windows 2003/XP is a distributed system connected via network media. It is assumed that the connectivity conditions described in Table 3-3 will exist.

Table 3-3 Connectivity Assumptions

Assumption	Description	PP Source
A.CONNECT	All connections to peripheral devices reside within the controlled access facilities. The TOE only addresses security concerns related to the manipulation of the TOE through its authorized access points. Internal communication paths to access points such as terminals are assumed to be adequately protected.	CAPP
A.PEER	Any other systems with which the TOE communicates are assumed to be under the same management control and operate under the same security policy constraints. The TOE is applicable to networked or distributed environments only if the entire network operates under the same constraints and resides within a single management domain. There are no security requirements that address the need to trust external systems or the communications links to such systems.	CAPP

3.3.2 Personnel Assumptions

It is assumed that the personnel conditions described in Table 3-4 will exist.

Table 3-4 Personnel Assumptions

Assumption	Description	PP Source
A.COOP	Authorized users possess the necessary authorization to access at least some of the information managed by the TOE and are expected to act in a cooperating manner in a benign environment.	CAPP
A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.	CAPP
A.NO_EVIL_ADM	The system administrative personnel are not careless, willfully negligent, or hostile, and will follow and abide by the instructions provided by the administrator documentation.	CAPP

3.3.3 Physical Assumptions

Windows 2003/XP is intended for application in user areas that have physical control and monitoring. It is assumed that the physical conditions described in Table 3-5 will exist.

Table 3-5 Physical Assumptions

Assumption	Description	PP Source
A.LOCATE	The processing resources of the TOE will be located within controlled access facilities that will prevent unauthorized physical access.	CAPP
A.PROTECT	The TOE hardware and software critical to security policy enforcement will be protected from unauthorized physical modification.	CAPP

4. Security Objectives

This section defines the security objectives of Windows 2003/XP and its supporting environment. Security objectives, categorized as either IT security objectives or non-IT security objectives, reflect the stated intent to counter identified threats and/or comply with any organizational security policies identified. All of the identified threats and organizational policies are addressed under one of the categories below.

4.1 TOE IT Security Objectives

Table 4-1 describes the Windows 2003/XP IT security objectives.

Table 4-1 IT Security Objectives

Security Objective	Description	PP Source
O.AUTHORIZATION	The TSF must ensure that only authorized users gain access to the TOE and its resources.	CAPP
O.DISCRETIONARY_ACCESS	The TSF must control accessed to resources based on identity of users. The TSF must allow authorized users to specify which resources may be accessed by which users.	CAPP
O.AUDITING	The TSF must record the security relevant actions of users of the TOE. The TSF must present this information to authorized administrators.	CAPP
O.RESIDUAL_INFORMATION	The TSF must ensure that any information contained in a protected resource is not released when the resource is recycled.	CAPP
O.MANAGE	The TSF must provide all the functions and facilities necessary to support the authorized administrators that are responsible for the management of TOE security.	CAPP
O.ENFORCEMENT	The TSF must be designed and implemented in a manner which ensures that the organizational policies are enforced in the target environment.	CAPP
O.AUDIT_PROTECTION	The TSF must provide the capability to protect audit information associated with individual users.	
O.PROTECT	The TSF must protect its own data and resources and must maintain a domain for its own execution that protects it from external interference or tampering.	
O.TRUSTED_PATH	The TSF must provide the capability to allow users to ensure they are not communicating with some other entity pretending to be the TSF during initial user authentication.	
O.LEGAL_WARNING	The TSF must provide a mechanism to advise users of legal issues involving use of the TOE prior to allowing the user to access resources controlled by the TSF.	
O.LIMIT_AUTHORIZATION	The TSF must provide the capability to limit the extent of each user's authorizations.	
O.IPSEC	The TSF must have the capability to protect data in transmission between distributed parts of the TOE and control the flow of traffic between distributed parts of the TOE	

Security Objective	Description	PP Source
O.ENCRYPTED_DATA	The TSF must ensure that only the users that encrypted data may receive that data decrypted.	
O.ASSURANCE	Assurance in the TOE's security functionality will be supported by the following activities: Configuration management of the TOE and its development evidence during its development; Use of sound design principles and techniques; Functional testing; demonstration that the guidance documentation is sufficient and not misleading; Vulnerability analysis; Penetration testing demonstrating the TOE is sufficiently robust to protect itself against the casual attacker using published exploits.	
O.MEDIATE	The TOE must mediate the flow of information between sets of TOE network interfaces or between a network interface and the TOE itself in accordance with its security policy.	

4.2 Non-IT Security Objectives for the 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 Non-IT Security Objectives for the Environment.

Table 4-2 Non-IT Security Objectives

Security Objective	Description	PP Source
O.INSTALL	Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner which maintains IT security objectives.	CAPP
O.PHYSICAL	Those responsible for the TOE must ensure that those parts of the TOE critical to security policy are protected from physical attack which might compromise IT security objectives.	CAPP
O.CREDEN	Those responsible for the TOE must ensure that all access credentials, such as passwords or other authentication information, are protected by the users in a manner which maintains IT security objectives.	CAPP

5. IT Security Requirements

5.1 TOE Security Functional Requirements

This section specifies the SFRs for the TOE. This section organizes the SFRs by CC class.

Requirement Operations:

Within the text of each SFR taken from the CAPP; assignment, refinement, and selection operations completed in the CAPP operations are underlined.

Within the text of each SFR taken from the CAPP, additional operations performed in this ST are identified as follows. Within the text of each SFR taken directly from the CC, operations performed in this ST are identified as follows:

- Additional selection and assignment operations completed in this ST are bracketed in this ST (e.g., [D).
- Additional refinement operations completed in this ST are indicated using bold, for additions, and strike-through, for deletions (e.g., "... **all** objects ..." or "... ~~some~~ **big** things ...").
- Additional iterations completed in this ST are italicized. Iterated requirements are indicated by a letter in parenthesis placed at the end of the component short name and element name(s) (e.g., FMT_MTD.1(a)).

Interpreted Requirements:

Requirements that have been modified based upon an International Interpretation are identified by an italicized parenthetical comment following the requirement element that has been modified (e.g., *(per International Interpretation #51)*).

SOF:

This ST includes the SOF assurance requirement (AVA_SOF.1). The minimum strength level for the SFRs realized by a probabilistic or permutational mechanism (with the exception of encryption mechanisms) is SOF-Medium.

SFR Summary:

Table 5-1, CAPP Components and Operations, summarizes the SFRs that are included in the ST from the CAPP as follows:

- Requirements included in the ST verbatim from the CAPP,
- Requirements operated upon in the CAPP,
- Requirements included in the ST with resolved operations from the CAPP, and
- Requirements supported by functions with associated SOF claims are identified with a "SOF" subscript in the column "CAPP Component."

Table 5-2, CC Components and Operations summarizes the SFRs that are not included in the CAPP as follows:

- Additional requirements from part 2 of the CC,
- Additional requirements from part 2 of the CC with resolved operations, and
- Requirements supported by functions with associated SOF claims are identified with a "SOF" subscript in the column "CC Component."

Explicitly Stated Requirements:

The CC envisioned that some PP/ST authors may have security needs not yet covered by the SFR components in the CC and allows PP/ST authors to consider requirements not taken from the CC, referred to as extensibility. This ST includes several requirements that are not derived from the CC. Some are inherited from the CAPP and others are not. Table 5-1 and 5-2 identifies those requirements that are not from the CC as those that have “extension” in the CAPP Operation or ST Operation Columns of those tables. These requirements are also denoted by their names ending with the phrase “_EX”.

All SFRs are fully stated in the sections below.

Table 5-1 CAPP Components and Operations

CAPP Component	Component Name	CAPP Operation	Additional ST Operations
FAU_GEN.1	Audit Data Generation	Assignment, Refinement	Refinement
FAU_GEN.2	User Identity Association	None	None
FAU_SAR.1	Audit Review	Assignment	Refinement
FAU_SAR.2	Restricted Audit Review	None	None
FAU_SAR.3(a)	Selectable Audit Review by Searching and Sorting	Assignment, Selection	Assignment, Selection, Iteration
FAU_SEL.1	Selective Audit	Selection	Assignment
FAU_STG.1	Protected Audit Trail Storage ²	Selection	Refinement
FAU_STG.3	Action in Case of Possible Audit Data Loss	Assignment	Assignment
FAU_STG.4	Prevention of Audit Data Loss	Selection	Assignment, Refinement
FDP_ACF.1(a)	Discretionary Access Control Functions	Assignment, Refinement	Assignment, Refinement, Iteration
FDP_RIP.2	Object Residual Information Protection	Selection	None
Note1_EX ³	Subject Residual Information Protection	Extension	None
FIA_ATD.1	User Attribute Definition	Assignment	Assignment
FIA_SOS.1 (SOF) ⁴	Verification of Secrets ⁵	Assignment	Refinement
FIA_UAU.1	Timing of Authentication	None	Assignment
FIA_UID.1	Timing of Identification	None	Assignment
FIA_UAU.7	Protected Authentication Feedback	Assignment	None

² This title is consistent with the CC. The CAPP title for this requirement is “Guarantees of Audit Data Availability” which is inconsistent with the CC.

³ This title is inconsistent with the CAPP in order to use this ST’s convention of denoting explicit requirements by ending the name with the phrase “_EX”. The CAPP titles this requirement as “FDP_RIP.2.Note1.”

⁴ The SOF claim associated with this requirement is a metric as defined in the FIA_SOS.1 requirement

⁵ This title is consistent with the CC. The CAPP title for this requirement is “Strength of Authentication Data” which is inconsistent with the CC.

CAPP Component	Component Name	CAPP Operation	Additional ST Operations
FIA_USB.1_EX ⁶	User-Subject Binding	Extension	Assignment, Refinement
FMT_MSA.1 (a)	Management of Object Security Attributes	Assignment, Selection	Assignment, Iteration
FMT_MSA.3(a)	Static Attribute Initialization	Assignment, Selection	Assignment, Iteration
FMT_MTD.1(a)	Management of the Audit Trail (1a)	Assignment, Selection, Iteration	None
FMT_MTD.1(b)	Management of Audited Events (1b)	Assignment, Selection, Iteration	None
FMT_MTD.1(c)	Management of User Attributes (1c)	Assignment, Selection, Iteration	Assignment
FMT_MTD.1(d)	Management of Authentication Data (1d)	Assignment, Selection, Iteration	None
FMT_REV.1(a)	Revocation of User Attributes (1a)	Assignment, Selection, Iteration	Assignment
FMT_REV.1(b)	Revocation of Object Attributes (1b)	Assignment, Selection, Iteration	Assignment, Refinement
FMT_SMR.1	Security Roles	Assignment	Assignment
FPT_AMT.1	Abstract Machine Testing	Selection	Refinement
FPT_RVM.1	Non-bypassability of the TSP ⁷	None	None
FPT_SEP.1	TSF Domain Separation	Assignment	None
FPT_STM.1	Reliable Time Stamps	None	None

Table 5-2 CC Components and Operations

CC Component	Component Name	ST Operations
FAU_SAR.3(b)	Selectable Audit Review by Searching	Assignment, Selection
FCS_COP.1(a) thru FCS_COP.1(j)	Cryptographic Operation	Assignment, Iteration

⁶ This title is inconsistent with the CAPP in order to use this ST's convention of denoting explicit requirements by ending the name with the phrase “_EX”. The CAPP titles this requirement as “FIA_USB.1”.

⁷ This title is consistent with the CC. The CAPP title for this requirement is “Reference Mediation” which is inconsistent with the CC.

CC Component	Component Name	ST Operations
FCS_CKM.1(a) thru FCS_CKM.1(e)	Cryptographic Key Generation	Assignment, Iteration, Refinement
FCS_CKM.2	Cryptographic Key Distribution	Assignment, Refinement
FCS_CKM.4	Cryptographic Key Zeroization	Refinement, Assignment
FCS_CKM_EX.1	Cryptographic Key Validation and Packaging	Extension
FCS_CKM_EX.2	Cryptographic Key Handling and Storage	Extension
FDP_ACC.2(a)	Discretionary Access Control Policy	Assignment, Refinement, Iteration
FDP_ACC.2(b)	WEBUSER Complete Access Control	Assignment, Refinement, Iteration
FDP_ACC.2(c)	Content-Provider Complete Access Control	Assignment, Refinement, Iteration
FDP_ACF.1(b)	WEBUSER Access Control Functions	Assignment, Refinement, Iteration
FDP_ACF.1(c)	Content-Provider Access Control Functions	Assignment, Refinement, Iteration
FDP_IFC.1(a)	IPSec Subset Information Flow Control	Assignment, Iteration
FDP_IFC.1(b)	Windows Firewall Connection Subset Information Flow Control	Assignment, Iteration
FDP_IFF.1(a)	IPSec Simple Security Attributes	Assignment, Refinement, Iteration
FDP_IFF.1(b)	Windows Firewall Connection Simple Security Attributes	Assignment, Refinement, Iteration
FDP_ITT.1	Basic Internal Protection	Assignment, Selection
FDP_UCT.1	WEBUSER SFP Basic Data Exchange Confidentiality	Assignment, Selection, Refinement
FDP_UIT.1	WEBUSER SFP Data Exchange Integrity	Assignment, Selection, Refinement

CC Component	Component Name	ST Operations
FIA_AFL.1	Authentication Failure Handling	Assignment
FIA_UAU.6	Re-authenticating	Refinement
FMT_MOF.1(a)	Management of Audit	Assignment, Selection, Iteration
FMT_MOF.1(b)	Management of TOE TSF Data in Transmission	Assignment, Selection, Iteration
FMT_MOF.1(c)	Management of Unlocking Sessions	Assignment, Selection, Iteration
FMT_MOF.1(d)	Management of Web Server	Assignment, Selection, Iteration, Refinement
FMT_MSA.1(b)	Management of DAC Object Security Attributes	Assignment, Selection, Iteration
FMT_MSA.1(c)	Management of IPSec Object Security Attributes	Assignment, Selection, Iteration
FMT_MSA.1(d)	Management of Windows Firewall Connection Object Security Attributes	Assignment, Selection, Iteration
FMT_MSA.1(e)	Management of WEBUSER Object Security Attributes	Assignment, Selection, Iteration
FMT_MSA.1(f)	Management of Content-Provider Object Security Attributes	Assignment, Selection, Iteration
FMT_MSA_EX.2	Valid Password Security Attributes	Extension
FMT_MSA.3(b)	IPSec Static Attribute Initialization	Assignment, Selection, Iteration
FMT_MSA.3(c)	Windows Firewall Connection Static Attribute Initialization	Assignment, Selection, Iteration
FMT_MSA.3(d)	WEBUSER Static Attribute Initialization	Assignment, Selection, Iteration, Refinement
FMT_MSA.3(e)	Content-Provider Static Attribute Initialization	Assignment, Selection, Iteration, Refinement
FMT_MTD.1(e)	Management of Account Lockout Duration	Assignment, Selection, Iteration
FMT_MTD.1(f)	Management of Minimum Password Length	Assignment, Selection, Iteration

CC Component	Component Name	ST Operations
FMT_MTD.1(g)	Management of TSF Time	Assignment, Selection, Iteration
FMT_MTD.1(h)	Management of NTFS Volume Quota Settings	Assignment, Selection, Iteration
FMT_MTD.1(i)	Management of Advisory Warning Message	Assignment, Selection, Iteration
FMT_MTD.1(j)	Management of Audit Log Size	Assignment, Selection, Iteration
FMT_MTD.1(k)	Management of User Inactivity Threshold	Assignment, Selection, Iteration
FMT_MTD.1(l)	Management of General TSF Data	Assignment, Selection, Iteration
FMT_MTD.1(m)	Management of Reading Authentication TSF Data	Assignment, Iteration, Refinement
FMT_MTD.1(n)	Management of Password Complexity Requirement	Assignment, Selection, Iteration
FMT_MTD.1(o)	Management of User Private/Public Key Pair	Assignment, Selection, Iteration
FMT_MTD.2	Management of Unsuccessful Authentication Attempts Threshold	Assignment
FMT_SAE.1	Timed-limited Authorization	Assignment
FMT-SMF.1	Specification of Management Functions	Assignment
FMT_SMR.3	Assuming Roles	Assignment, Refinement
TRANSFER_PROT_EX.1	Internal TSF Data Transfer Protection	Extension
FPT_TRC_EX	Internal TSF Data Consistency	Extension
TRANSFER_PROT_EX.3	Internal TSF Data Integrity Monitoring	Extension
FPT_RPL_EX.1	Replay Detection	Extension
FRU_RSA.1	Maximum Quotas	Assignment, Selection
FTA_LSA_EX.1	Limitation on Scope of Selectable Attributes	Extension
FTA_MCS_EX.1	Basic limitation on multiple concurrent sessions	Extension
FTA_SSL1	TSF-initiated Session Locking	Assignment
FTA_SSL.2	User-initiated Session Locking	Assignment

CC Component	Component Name	ST Operations
FTA_SSL.3	WEBUSER TSF-Initiated Termination	Assignment, Refinement
FTA_TAB.1	Default TOE Access Banners	Refinement
FTA_TSE.1	TOE Session Establishment	Assignment
FTP_TRP.1	Trusted Path	Assignment, Selection

5.1.1 Security Audit (FAU) Requirements

5.1.1.1 Audit Data Generation (FAU_GEN.1)

5.1.1.1.1 FAU_GEN.1.1

The TSF shall be able to generate an audit record of the auditable events listed in column “Event” of **Table 5-3 (CAPP Compliant Auditable Events)** and the events listed in column “Event” of **Table 5-4 (Other Auditable Events)**.

5.1.1.1.2 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, and the outcome (success or failure) of the event; and
- b) The additional information specified in the “Details” column of **Table 5-3, CAPP Compliant Auditable Events**. **This includes:**
 - **The auditable events associated with the CAPP SFRs at the basic level of auditing, except FIA_UID’s user identity during failures**
 - **The identified auditable events associated with SFRs in this ST, which are not included in the CAPP, at the not specified level of audit**

Table 5-3 CAPP Compliant Auditable Events

Component	Event	Details
FAU_GEN.1	Start-up and Shutdown of the audit functions	
FAU_GEN.2	None	
FAU_SAR.1	Reading of information from the audit records	
FAU_SAR.2	Unsuccessful attempts to read information from the audit records	
FAU_SAR.3(a), (b)	None	
FAU_STG.1	None	
FAU_STG.3	Actions taken due to exceeding of a threshold	

Component	Event	Details
FAU_STG.4	Actions taken due to the audit storage failure	
FDP_ACC.1 (a) ⁸	None	
FDP_ACF.1 (a)	All requests to perform an operation on an object covered by the SFP	The identity of the object.
FDP_RIP.2	None	
FDP_RIP.2. Note 1	None	
FIA_ATD.1	None	
FIA_SOS.1	Rejection or acceptance by the TSF of any tested secret	
FIA_UAU.1	The use of the authentication mechanism	
FIA_UAU.7	None	
FIA_UID.1	All use of the user identification mechanism, including the identity provided during successful attempts	The origin of the attempt (e.g. terminal identification).
FIA_USB.1_EX	Success and failure of binding user security attributes to a subject (e.g., success and failure to create a subject)	
FMT_MSA.1(a)	All modifications of the values of security attributes	
FMT_MSA.3(a)	Modifications of the default setting of permissive or restrictive rules. All modifications of the initial value of security attributes.	
FMT_MTD.1(a) CAPP – 5.4.3	All modifications to the values of TSF data	
FMT_MTD.1(b) CAPP – 5.4.4	All modifications to the values of TSF data	The new value of the TSF data.
FMT_MTD.1(c) CAPP – 5.4.5	All modifications to the values of TSF data	The new value of the TSF data.
FMT_MTD.1(d) CAPP- 5.4.6	All modifications to the values of TSF data	
FMT_REV.1(a) CAPP – 5.4.7	All attempts to revoke security attributes	
FMT_REV.1(b) CAPP – 5.4.8	All modifications to the values of TSF data	
FMT_SMR.1	Modifications to the group of users that are part of a role	
FMT_SMR.1	Every use of the rights of a role. (Additional/ Detailed)	The role and the origin of the request.
FPT_RVM.1	None	
FPT_SEP.1	None	
FPT_STM.1	Changes to the time	

⁸ This requirement is not included in this ST, however, FDP_ACC.2 is which is hierarchical to FDP_ACC.1.

Table 5-4 Other Auditable Events

Component	Event
FIA_AFL.1	Account locked out due to exceeding the maximum number of unsuccessful logon attempts
FMT_MOF.1(a)	Audit Policy Changes
FMT_MTD.1(g)	Attempt to use an authorized administrator privilege to change the TSF Time
TRANSFER_PROT_EX.1	IPSEC policy changes
FTA_SSL1	Attempt to unlock
FTA_SSL.2	Attempt to unlock
FTA_TSE.1	Logon Failure due to password expiration
FTP_TRP.1	Authentication and unlocking attempts
FMT_MTD.1(e)	Lockout Duration changes
FMT_MTD.1(f)	Modification of minimum password length
FMT_MTD.1(n)	Modification of password complexity policy
FMT_MTD.2	Modification of unsuccessful logon attempt threshold
FMT_SAE.1	Setting of password expiration time
TRANSFER_PROT_EX.3	Detection of a data integrity violation
FPT_RPL.1	Replay of TSF data
FPT_TRC_EX.1	Directory Replication

5.1.1.2 User Identity Association (FAU_GEN.2)

5.1.1.2.1 FAU_GEN.2.1

The TSF shall be able to associate each auditable event with the identity of the user that caused the event.

5.1.1.3 Audit Review (FAU_SAR.1)

5.1.1.3.1 FAU_SAR.1.1

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

5.1.1.3.2 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 trail**.

5.1.1.4 Restricted Audit Review (FAU_SAR.2)

5.1.1.4.1 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.

5.1.1.5 Selectable Audit Review by Searching and Sorting (FAU_SAR.3(a))

5.1.1.5.1 FAU_SAR.3.1(a)

The TSF shall provide the ability to perform [searches and sorting] of audit data based on the following attributes:

a) User identity:

b) [Type (success and/or failure), date, time, category, event identifier, and computer].

5.1.1.6 Selectable Audit Review by Searching (FAU_SAR.3(b))

5.1.1.6.1 FAU_SAR.3.1(b)

The TSF shall provide the ability to perform [searches] of audit data based on [free form text substring within audit records].

5.1.1.7 Selective Audit (FAU_SEL.1)

5.1.1.7.1 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) User identity:

b) [Object identity, Host identity, Event type, Success of auditable security events, and Failure of auditable security events.]

5.1.1.8 Protected Audit Trail Storage (FAU_STG.1)

5.1.1.8.1 FAU_STG.1.1

The TSF shall protect the stored audit records from unauthorized deletion.

5.1.1.8.2 FAU_STG.1.2

The TSF shall be able to prevent ~~unauthorized~~ modifications to the audit records in the audit trail. (*per International Interpretation #141*)

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

5.1.1.9.1 FAU_STG.3.1

The TSF shall generate an alarm to the authorized administrator if the audit trail exceeds [the authorized administrator specified log size].

5.1.1.10 Prevention of Audit Data Loss (FAU_STG.4)

5.1.1.10.1 FAU_STG.4.1

When the audit trail becomes full, the TSF shall ~~be able to~~ **provide the authorized administrator the capability to prevent auditable events, except those taken by the authorized administrator (in the context of performing TOE maintenance)** and [generate an alarm to the authorized administrator]. ~~if the audit trail is full~~

5.1.2 Cryptographic Support (FCS)

5.1.2.1 Cryptographic Operation (DES Encryption and Decryption) (FCS_COP.1(a))

5.1.2.1.1 FCS_COP.1.1(a)

The TSF shall perform [the TOE CryptProtectData, CryptEncrypt, FipsDes, and Fips3Des3Key encryption function and the CryptUnprotectData, CryptDecrypt, FipsDes, and Fips3Des3Key decryption function] in accordance with a specified cryptographic algorithm [FIPS-46-3 DES or Triple DES ECB and CBC modes] and cryptographic key sizes [56-bits or 168-bits respectively] that meet the following [FIPS 46-3].

5.1.2.2 Cryptographic Operation (RSA Hash) (FCS_COP.1(b))

5.1.2.2.1 FCS_COP.1.1(b)

The TSF shall perform [the TOE CryptSignHash signing function] in accordance with a specified cryptographic algorithm [FIPS-186-2 RSA using PKCS-1 (RSASSA-PKCS1-v1_5)] and cryptographic key size [default 1024 bits and maximum 16384 bits] that meets the following [none].

5.1.2.3 Cryptographic Operation (DSA Hash) (FCS_COP.1(c))

5.1.2.3.1 FCS_COP.1.1(c)

The TSF shall perform [the TOE CryptSignHash signing function] in accordance with a specified cryptographic algorithm [FIPS-186-2 DSA] and cryptographic key size [default 1024 bits] that meets the following [none].

5.1.2.4 Cryptographic Operation (XP SHA Hash) (FCS_COP.1(d))

5.1.2.4.1 FCS_COP.1.1(d)

The TSF shall perform [the TOE CryptCreateHash, CryptHashData, and CryptGetHashParam hashing functions with the Algid parameter CALG_SHA1] in accordance with a specified cryptographic algorithm [FIPS-180-2 SHA-1] and cryptographic key size [not applicable] that meet the following [FIPS 180-2].

5.1.2.5 Cryptographic Operation (Server SHA Hash) (FCS_COP.1(e))

5.1.2.5.1 FCS_COP.1.1(e)

The TSF shall perform [the TOE CryptCreateHash, CryptHashData, and CryptGetHashParam hashing functions with the Algid parameter CALG_SHA1, CALG_SHA_256, CALG_SHA_384, or CALG_SHA_512 hashing] in accordance with a specified cryptographic algorithm [FIPS-180-2 SHA-1, SHA-256, SHA-384, and SHA-512 respectively] and cryptographic key size [not applicable] that meet the following [FIPS-180-2, SHA-1, SHA-256, SHA-384, and SHA-512].

5.1.2.6 Cryptographic Operation (Random Number Generator) (FCS_COP.1(f))

5.1.2.6.1 FCS_COP.1.1(f)

The TSF shall perform [the TOE *CryptGenRandom* and *FIPSGenRandom* pseudo random number generation functions] in accordance with a specified cryptographic algorithm [SHA-1] and cryptographic key size [not applicable] that meet the following [none].

5.1.2.7 Cryptographic Operation (RSA Encryption and Decryption) (FCS_COP.1(g))

5.1.2.7.1 FCS_COP.1.1(g)

The TSF shall perform [the TOE *CryptEncrypt* and *CryptDecrypt* public key encryption services] in accordance with a specified cryptographic algorithm [RSA] and cryptographic key size [default 1024 bits and maximum 16384 bits] that meet the following [PKCS#1(RSAES-PKCS1-v1_5)].

5.1.2.8 Cryptographic Operation (Diffie-Hellman) (FCS_COP.1(h))

5.1.2.8.1 FCS_COP.1.1(h)

The TSF shall perform [the TOE *CryptExportKey*, *CryptImportKey* and *CryptGetKeyParam*, *CryptGenKey* cryptographic key establishment services] in accordance with a specified cryptographic algorithm [Diffie-Hellman (Ephemeral-Ephemeral)] and cryptographic key size [default 1024 bits and maximum 2048 bits] that meet the following [none].

5.1.2.9 Cryptographic Operation (AES Encryption and Decryption) (FCS_COP.1(i))

5.1.2.9.1 FCS_COP.1.1(i)

The TSF shall perform [the TOE *CryptEncrypt* encryption function and the *CryptDecrypt* decryption function] in accordance with a specified cryptographic algorithm [AES ECB and CBC modes] and cryptographic key sizes [128 bits, 192 bits, or 256 bits] that meet the following [FIPS-197].

5.1.2.10 Cryptographic Operation (HMAC) (FCS_COP.1(j))

5.1.2.10.1 FCS_COP.1.1(j)

The TSF shall perform [the TOE *CryptCreateHash*, *CryptHashData*, and *CryptGetHashParam* hashing functions] in accordance with a specified cryptographic algorithm [HMAC] and cryptographic key sizes [128] that meet the following [FIPS-198 (SHA-1)].

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

5.1.2.11.1 FCS_CKM.1.1(a)

The TSF shall generate **FIPS-46-3 DES or Triple DES symmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm **as follows:**

- [
- (1) a software random number generator (RNG) as specified in FCS_COP.1.1(f), or
 - (2) a key establishment scheme as specified in FCS_COP.1.1(h) based on public key cryptography using a software random number generator (RNG) as specified in FCS_COP.1.1(f),

when using the TOE *CryptGenKey* function, or when inside the TOE EFS Service, DPAPI Service, Schannel Security Package, or IPSEC Service,

]

and specified cryptographic key sizes [56 bits or 168 bits respectively] that meet the following [FIPS 140-1 or 140-2 Level 1].

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

5.1.2.12.1 FCS_CKM.1.1(b)

The TSF shall generate **FIPS-186-2/PKCS#1 asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm **as follows**:

- [
- (1) a prime number generator, when using the TOE CryptGenKey function,
-]

and specified cryptographic key sizes [default 1024 bits and maximum 16384] that meets the following [none].

5.1.2.13 Cryptographic Key Management (FCS_CKM) Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1(c))

5.1.2.13.1 FCS_CKM.1.1(c)

The TSF shall generate **FIPS-186-2 DSA asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm **as follows**:

- [
- (1) a random number generator (SHA-1 based RNG), when using the TOE CryptGenKey function,
-]

and specified cryptographic key sizes [1024 bits] that meets the following [FIPS 186-2].

5.1.2.14 Cryptographic Key Management (FCS_CKM) Cryptographic Key Generation (for symmetric keys) (FCS_CKM.1(d))

5.1.2.14.1 FCS_CKM.1.1(d)

The TSF shall generate **FIPS-197AES symmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm **as follows**:

- [
- (1) a software random number generator (RNG) as specified in FCS_COP.1.1(f), or
 - (2) a key establishment scheme as specified in FCS_COP.1.1(h) based on public key cryptography using a software random number generator (RNG) as specified in FCS_COP.1.1(f),

when using the TOE CryptGenKey function,

] and specified cryptographic key sizes [128 bits, 192 bits, or 256 bits] that meet the following [**FIPS-197AES symmetric**].

5.1.2.15 Cryptographic Key Management (FCS_CKM) Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1(e))

5.1.2.15.1 FCS_CKM.1.1(e)

The TSF shall generate **Diffie-Hellman asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm **as follows**:

- [
- (1) a random number generator (SHA-1 based RNG), when using the TOE CryptGenKey function,
-]

and specified cryptographic key sizes [default 1024 bits and maximum 2048 bits] that meets the following [none].

5.1.2.16 Cryptographic Key Management (FCS_CKM) Cryptographic Key Distribution (FCS_CKM.2)

5.1.2.16.1 FCS_CKM.2.1

The TSF shall distribute **FIPS-186-2 DSA and FIPS-186-2/PKCS#1 private asymmetric** cryptographic keys in accordance with a specified cryptographic key distribution method **as follows:** [

- (1) PKCS#5 V2.0 (Password-Based Encryption Standard), inside the TOE DPAPI service for handling the distribution of the aforementioned keys to physically-separated parts of the TOE that the keys owners are currently logging onto and needing to use the keys, using FIPS-180-2 SHA-1 and FIPS-46-3 Triple DES,

]
that meet the following [FIPS 140-1 or 140-2 Level 1].

5.1.2.17 Cryptographic Key Management (FCS_CKM) Cryptographic Key Zeroization (FCS_CKM.4)

5.1.2.17.1 FCS_CKM.4.1

The TSF shall destroy cryptographic keys **within the FIPS-140 validated cryptographic modules** in accordance with a specified cryptographic key destruction method [cryptographic key zeroization method] that meets the following [FIPS 140-1 or 140-2 Level 1].

5.1.2.18 Cryptographic Key Management (FCS_CKM) Cryptographic Key Validation and Packaging (FCS_CKM_EX.1)

5.1.2.18.1 FCS_CKM_EX.1.1

The TSF shall apply validation techniques (e.g. hash) to validate the CryptProtectData-encrypted FIPS-186-2 DSA and FIPS-186-2/PKCS#1 private asymmetric cryptographic keys when they are obtained via the distribution method as called out in FCS_CKM.2.1 or via the storage method as called out in FCS_CKM_EXP.2.2.

5.1.2.19 Cryptographic Key Management (FCS_CKM) Cryptographic Key Handling and Storage (FCS_CKM_EX.2)

5.1.2.19.1 FCS_CKM_EX.2.1

The TSF shall provide a means to ensure that FIPS-186-2 DSA and FIPS-186-2/PKCS#1 private asymmetric cryptographic keys are associated with the correct entities (i.e., person) to which the keys are assigned.

5.1.2.19.2 FCS_CKM_EX.2.2

The TSF shall store FIPS-186-2 DSA and FIPS-186-2/PKCS#1 private asymmetric cryptographic keys in accordance with a specified cryptographic key storage method as follows: [

- (1) PKCS#5 V2.0 (Password-Based Encryption Standard), using FIPS-180-2 SHA-1 and FIPS-46-3 Triple DES,

]
that meet the following [FIPS 140-1 or 140-2 Level 1].

5.1.3 User Data Protection (FDP) Requirements

5.1.3.1 Discretionary Access Control Policy (FDP_ACC.2 (a))

5.1.3.1.1 FDP_ACC.2.1(a)

The TSF shall enforce the [Discretionary Access Control Policy] on
[subjects – processes acting on the behalf of users]
and

[Named objects – Desktop, Event, Keyed Event, Event pair, I/O Completion Port, Job, Key, Mutant, Mailslot, Named pipe, NTFS directory, NTFS file, Object Directory, LPC Port, Printer, Process, Section, Semaphore, Symbolic Link, Thread, Timer, Tokens, Volume, Window Station, Active Directory, Application Pool File, URL Reservation, debug, Filter Communication Port, and Filter Connection Port objects]; and all operations among ~~them~~ **subjects and objects covered by the SFP.**

5.1.3.1.2 FDP_ACC.2.2(a)

The TSF shall ensure that all operations between any subject ~~in the TSC~~ and any **named** object ~~within the TSC~~ are covered by ~~an access control SFP~~ **the Discretionary Access Control policy.**

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

5.1.3.2.1 FDP_ACC.2.1(b)

*The **Web Server part of the TSF** shall enforce the [WEBUSER SFP] on*

[

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.

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.

5.1.3.2.2 FDP_ACC.2.2(b)

*The TSF shall ensure that all operations between any subject in the **WEBUSER** TSC and any object within the **WEBUSER** TSC are covered by ~~an access control~~ **the WEBUSER SFP.***

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

5.1.3.3.1 FDP_ACC.2.1(c)

*The **Web Server part of the TSF** shall enforce the [CONTENT-PROVIDER (CP) SFP] on*

[

subjects - 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)

objects - Web Server Content (served by the Web Server part of TSF over http:// or https://)

]

and upon all operations among **Web Server** subjects and **Web server** objects covered by the **CONTENT-PROVIDER SFP**:

5.1.3.3.2 FDP_ACC.2.2(c)

The **Web Server part of the TSF** shall ensure that all operations between any subject in the **CONTENT-PROVIDER TSC** and any object within the **CONTENT-PROVIDER TSC** are covered by ~~an access control~~ the **CONTENT-PROVIDER SFP**.

5.1.3.4 Discretionary Access Control Functions (FDP_ACF.1(a))

5.1.3.4.1 FDP_ACF.1.1(a)

The TSF shall enforce the Discretionary Access Control Policy to objects based on the following:

- a) The user identity, group membership(s), and privileges associated with a subject
- b) **The user private key (only applicable when requesting access to encrypted files or sign data hashes with the TOE CryptSignHash function) associated with a subject**
- c) The following access control attributes associated with an object:

[

- Object Owner
- A Discretionary Access Control List (DACL) that can be either absent, empty, or consist of a list of one or more entries. Each DACL entry has a:
 - Type (allow or deny)
 - User or group identifier
 - Specific object access right bitmasks
 - For directory service (DS) object entries, a globally unique identifier (GUID) indicating a DS-specific object attribute.
- For encrypted file objects, File Encryption Keys (FEKs)

The defaults for allowed or denied operations are:

- If a DACL is absent, the object is not protected and all access is granted.
- If a DACL is present but empty, no access is granted.

]. (*per International Interpretation #103*)

5.1.3.4.2 FDP_ACF.1.2(a)

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

[Object access is allowed if at least one of the following conditions is true:

- A DACL entry explicitly grants access to a user, and the access has not been denied by a previous entry in the DACL.
- A DACL entry explicitly grants access to a Group of which the subject is a member, and the access has not been denied by a previous entry in the DACL
- A DACL is not present
- The subject is the object owner and the operation is to view the object's DACL or to modify the object's DACL

]

5.1.3.4.3 FDP_ACF.1.3(a)

The TSF shall explicitly authorize access of subjects to objects based in the following additional rules:

[

- For the following operation, the authorized administrator can bypass the rules listed in FDP_ACF.1.2:
Request to change the owner of an object
- For the following operations, only the authorized administrator can be granted access and the rules in FDP_ACF.1.2 do not apply:
Request to change or modify the auditing of access attempts to an object
- For encrypted file objects, in addition to meeting FDP_ACF.1.2, the user must have a private key that can decrypt the FEK associated with the file.

].

5.1.3.4.4 FDP_ACF.1.4(a)

The TSF shall explicitly deny access of subjects to objects based on the following rules:

[

Object access is explicitly denied if at least one of the below conditions is true:

- A DACL entry explicitly denies access for a user, and the access has not been granted by a previous entry in the DACL.
- A DACL entry explicitly denies access for the group of which the user is a member, and the access has not been granted by a previous entry in the DACL.

].

5.1.3.5 WEBUSER Access Control Functions (FDP_ACF.1(b))

5.1.3.5.1 FDP_ACF.1.1(b)

The Web Server part of the TSF shall enforce the [WEBUSER SFP] to controlled-access content objects based on the following types of subject and object security attributes:

[

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

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

objects – Web Server objects – web server content (served by the Web Server part of the TSF over http:// or https://)

- *the DACL associated with the object*
- *the web permissions associated with an object*
- *the URL authorization associated with an object.*

]

5.1.3.5.2 FDP_ACF.1.2(b)

The Web Server part of the TSF shall enforce the following WEBUSER SFP ordered rules to determine if an operation among controlled subjects and controlled objects is allowed:

[

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

]

5.1.3.5.3 FDP_ACF.1.3(b)

The **Web Server part of the TSF** shall explicitly authorize access of **Web Server** subjects to **Web Server** objects based on the following additional **WEBUSER SFP** rules:

[

- (a) *a web user trying to access an object URL must be authorized to the operation AccessURL, 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.*

]

5.1.3.5.4 FDP_ACF.1.4(b)

The **Web Server part of the TSF** shall explicitly deny access of **Web Server** subjects to **Web Server** objects based on the following additional **WEBUSER SFP** rules:

[

- (a) *if a web user uses http:// instead of https:// and the web permission associated with the object requires SSL.*
- (b) *if a web user does not use a client certificate and the web permission associated with the object requires 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 SSL and 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.1.3)*

]

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

5.1.3.6.1 FDP_ACF.1.1(c)

The **Web Server part of the TSF** shall enforce the [CONTENT-PROVIDER SFP] to objects based on the following **types of subject and object security attributes**:

[

subjects – 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)

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

objects: Web Server Content (served by the Web Server part of the TSF over http:// or https://

- *the web permissions associated with an object*
- *the DACL associated with the object*
- *the URL authorization.*

]

5.1.3.6.2 FDP_ACF.1.2(c)

The **Web Server part of the TSF** shall enforce the following **CONTENT-PROVIDER SFP** 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:*
 1. *If the requested access is denied by the file's DACL associated with the web content to that web user, deny access.*
 2. *If the requested access is something other than read access, deny access.*
 3. *If read-only access is permitted to that authorized web user by the file's DACL associated with the web content, grant access*
 4. *Otherwise, deny access.*
- (c) *For (Web Server) public content*
 1. *If the requested access is something other than read access, deny access.*
 2. *Grant read-only access to web user.*

]

5.1.3.6.3 FDP_ACF.1.3(c)

The **Web Server part of the TSF** shall explicitly authorize access of subjects to objects based on the following additional **CONTENT-PROVIDER SFP** rules:

[

- (a) *a content provider trying to access an object URL must be authorized to the operation AccessURL 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 access the source of web server content if the web permission associated with the object allows access to the source*
- (e) *a content provider may view web server content file lists and collections if the web permission associated with the object allows browsing access*

]

5.1.3.6.4 FDP_ACF.1.4(c)

The **Web Server part of the TSF** shall explicitly deny access of subjects to objects based on the following additional **CONTENT-PROVIDER SFP** rules:

[

- (a) *if a content provider uses http:// instead of https:// and the web permission associated with the object requires SSL.*
- (b) *if a content provider does not use a client certificate and the web permission associated with the object requires 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 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.1.3(c))*

]

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

5.1.3.7.1 FDP_IFC.1.1(a)

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;*
- (2) *information: IP traffic*
- (3) *operation: pass information.*

].

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

5.1.3.8.1 FDP_IFC.1.1(b)

The TSF shall enforce the [Connection Firewall Policy] on:

[

- a) *subjects: one TSF receiving IP traffic from another TSF;*
- a) *information: IP traffic*
- b) *operation: receive information.*

].

5.1.3.9 IPSec Simple Security Attributes (FDP_IFF.1(a))

5.1.3.9.1 FDP_IFF.1.1(a)

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
- destination port identification .

]. (per International Interpretation #104)

5.1.3.9.2 FDP_IFF.1.2(a)

The TSF shall permit an information flow between a controlled subject and **another** controlled ~~information~~ **subject** 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]

5.1.3.9.3 FDP_IFF.1.3(a)

The TSF shall enforce the [no additional information control SFP rules].

5.1.3.9.4 FDP_IFF.1.4(a)

The TSF shall provide the following [no additional SFP capabilities].

5.1.3.9.5 FDP_IFF.1.5(a)

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

5.1.3.9.6 FDP_IFF.1.6(a)

The TSF shall explicitly deny an information flow based on the following rules: [no explicit deny rules].

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

5.1.3.10.1 FDP_IFF.1.1(b)

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 .*

]. (*per International Interpretation #104*)

5.1.3.10.2 FDP_IFF.1.2(b)

The TSF shall permit an information flow between a controlled subject and **another** controlled ~~information~~ **subject** via a controlled operation if the following rules hold:

[the incoming packet is a response to previous outgoing packet]

5.1.3.10.3 FDP_IFF.1.3(b)

The TSF shall enforce the *[no additional information control SFP rules]*.

5.1.3.10.4 FDP_IFF.1.4(b)

The TSF shall provide the following *[no additional SFP capabilities]*.

5.1.3.10.5 FDP_IFF.1.5(b)

The TSF shall explicitly authorize an information flow based on the following rules: *[the destination port is permitted by the Windows Firewall Connection Policy Port Mapping Rules]*.

5.1.3.10.6 FDP_IFF.1.6(b)

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

5.1.3.11 Basic Internal Transfer Protection (FDP_ITT.1)

5.1.3.11.1 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.

5.1.3.12 Object Residual Information Protection (FDP_RIP.2)

5.1.3.12.1 FDP_RIP.2.1

The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to all objects.

5.1.3.13 WEBUSER Basic Data Exchange Confidentiality (FDP_UCT.1)

5.1.3.13.1 FDP_UCT.1.1

The **Web Server part of the** TSF shall enforce the [WEBUSER SFP] to be able to [transmit and receive] **Web Server controlled-access content** objects in a manner protected from unauthorized disclosure

5.1.3.14 WEBUSER SFP Data Exchange Integrity (FDP_UIT.1)

5.1.3.14.1 FDP_UIT.1.1

The **Web Server part of the** TSF shall enforce the [WEBUSER SFP] to be able to [transmit and receive] **Web Server controlled-access content** ~~user~~ data in a manner protected from [modification] errors.

5.1.3.14.2 FDP_UIT.1.2

The **Web Server part of the** TSF shall be able to determine on receipt of **Web Server controlled-access content** ~~user~~ data, **under the WEBUSER SFP**, whether [modification] has occurred.

5.1.3.15 Subject Residual Information Protection (Note1_EX)

5.1.3.15.1 Note1_EX.1

The TSF shall ensure that any previous information content of a resource is made unavailable upon the allocation of the resource to all subjects.

5.1.4 Identification and Authentication (FIA)

5.1.3.1 Authentication Failure Handling (FIA_AFL.1)

5.1.4.1.1 FIA_AFL.1.1

The TSF shall detect when [an administrator configurable positive integer within a range of values acceptable to the administrator] unsuccessful authentication attempts occur related to [user logon].

5.1.4.1.2 FIA_AFL.1.2

When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall [disable the user account for an authorized administrator specified duration].

5.1.4.2 User Attribute Definition (FIA_ATD.1)

5.1.4.2.1 FIA_ATD.1.1

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

- a) User Identifier;
- b) Group Memberships;
- c) Authentication Data;
- d) Security-relevant Roles; and
- e) [Private/Public Keys, Privileges, and Logon Rights on specific physically separated parts of the TOE; Allowable time and day to logon; Policy requiring smart card to logon]

5.1.4.3 Verification of Secrets (FIA_SOS.1)

5.1.4.3.1 FIA_SOS.1.1

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

- a) For each attempt to use the authentication mechanism, the probability that a random attempt will succeed is less than one in 4,000,000 2×10^{15} ;

- b) For multiple attempts to use the authentication mechanism during a one minute period, the probability that a random attempt during that minute will succeed is less than one in ~~100,000~~ 25,000,000,000,000;
- c) Any feedback given during an attempt to use the authentication mechanism will not reduce the probability below the above metrics; and
- d) **The authentication mechanism must provide a delay between attempts, such that there can be no more than ten attempts per minute.**

5.1.4.4 Timing of Authentication (FIA_UAU.1)

5.1.4.4.1 FIA_UAU.1.1

The TSF shall allow [access to the web server] on behalf of that user to be performed before the user is authenticated.

5.1.4.4.2 FIA_UAU.1.2

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

5.1.4.5 Re-authenticating (FIA_UAU.6)

5.1.4.5.1 FIA_UAU.6.1

The TSF shall re-authenticate the user ~~under the conditions [assignment: list of conditions under which re-authentication is required]~~ **when changing authentication data.**

5.1.4.6 Protected Authentication Feedback (FIA_UAU.7)

5.1.4.6.1 FIA_UAU.7.1

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

5.1.4.7 Timing of Identification (FIA_UID.1)

5.1.4.7.1 FIA_UID.1.1

The TSF shall allow [access to the web server] on behalf of that user.

5.1.4.7.2 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.

5.1.4.8 User Subject Binding (FIA_USB.1_EX)

5.1.4.8.1 FIA_USB.1_EX.1

The TSF shall associate the following user security attributes with subjects acting on the behalf of that user:

- a) The user **unique** identity which is associated with auditable events;
- b) The user identity or identities which are used to enforce the Discretionary Access Control Policy, and Maximum Quotas (FRU_RSA.1);

- c) The group membership or memberships used to enforce the Discretionary Access Control Policy:
- d) [Private/Public Keys, Privileges.]

5.1.4.8.2 FIA_USB.1_EX.2

The TSF shall enforce the following rules on the initial association of user security attributes with subjects acting on the behalf of a user:

- a) [Every subject will be assigned a subset of security attributes associated with the user on whose behalf the subject will act.]

5.1.4.8.3 FIA_USB.1_EX.3

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

- a) [Subjects acting on behalf of users cannot add additional security attributes beyond those initially assigned.]

5.1.5 Management Requirements (FMT)

5.1.5.1 Management of Audit (FMT_MOF.1(a))

5.1.5.1.1 FMT_MOF.1.1(a)

The TSF shall restrict the ability to [enable, disable, modify the behavior of] the function [audit] to [authorized administrators].

5.1.5.2 Management of TOE TSF Data in Transmission (FMT_MOF.1(b))

5.1.5.2.1 FMT_MOF.1.1(b)

The TSF shall restrict the ability to [determine the behavior of and modify the behavior of] the function [that protect TOE Data during transmission between separate parts of the TOE] to [authorized administrators].

5.1.5.3 Management of Unlocking Sessions (FMT_MOF.1(c))

5.1.5.3.1 FMT_MOF.1.1(c)

The TSF shall restrict the ability to [modify the behavior of] the function [locked user session] to [authorized administrators and authorized user of locked session].

5.1.5.4 Management of the Web Server (FMT_MOF.1(d))

5.1.5.4.1 FMT_MOF.1.1(d)

*The **Web Server part of the** TSF shall restrict the ability to [modify the behaviour of] the function [WEBUSER SFP] to [authorized administrators].*

5.1.5.5 Management of Object Security Attributes (FMT_MSA.1(a))

5.1.5.5.1 FMT_MSA.1.1(a)

The TSF shall enforce the Discretionary Access Control Policy to restrict the ability to modify the security attributes access control attributes associated with a named object to [the owner of the object, subjects with DAC permission to take ownership or to modify the DACL, and subjects with a specific privilege].

5.1.5.6 Management of DAC Object Security Attributes (FMT_MSA.1(b))

5.1.5.6.1 FMT_MSA.1.1(b)

The TSF shall enforce the [Discretionary Access Control Policy] to restrict the ability to [delete] the security attributes [File Encryption Keys (FEKs)] to [users with access to one of the private keys used to protect the file encryption key associated with the file and subjects with a specific privilege].

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

5.1.5.7.1 FMT_MSA.1.1(c)

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].

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

5.1.5.8.1 FMT_MSA.1.1(d)

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].

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

5.1.5.9.1 FMT_MSA.1.1(e)

The TSF shall enforce the [WEBUSER Policy] to restrict the ability to [modify] the security attributes [WEBUSER Policy security attributes] to [the authorized administrator].

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

5.1.5.10.1 FMT_MSA.1.1(f)

The TSF shall enforce the [CONTENT-PROVIDER Policy] to restrict the ability to [modify] the security attributes [CONTENT-PROVIDER Policy security attributes] to [the authorized administrator].

5.1.5.11 Valid Password Security Attributes (FMT_MSA_EX.2)

5.1.5.11.1 FMT_MSA_EX.2.1

The TSF shall ensure that only values meeting the password complexity restrictions, if defined by the authorized administrator, are accepted for password security attributes.

5.1.5.12 Static Attribute Initialization (FMT_MSA.3(a))

5.1.5.12.1 FMT_MSA.3.1(a)

The TSF shall enforce the Discretionary Access Control Policy to provide restrictive default values for security attributes that are used to enforce the Discretionary Access Control Policy.

5.1.5.12.2 FMT_MSA.3.2(a)

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

5.1.5.13 IPsec Static Attribute Initialization (FMT_MSA.3(b))

5.1.5.13.1 FMT_MSA.3.1(b)

The TSF shall enforce the [IPsec Filter Policy] to provide [permissive] default values for security attributes that are used to enforce the SFP.

5.1.5.13.2 FMT_MSA.3.2(b)

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.

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

5.1.5.14.1 FMT_MSA.3.1(c)

The TSF shall enforce the [Windows Firewall Connection Policy] to provide [permissive] default values for security attributes that are used to enforce the SFP.

5.1.5.14.2 FMT_MSA.3.2(c)

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

5.1.5.15 WEBUSER Static Attribute Initialization (FMT_MSA.3(d))

5.1.5.15.1 FMT_MSA.3.1(d)

*The **Web Server part of the TSF** shall enforce the [WEBUSER SFP] to provide [restrictive] default values for security attributes that are used to enforce the SFP.*

5.1.5.15.2 FMT_MSA.3.2(d)

*The **Web Server part of the TSF** shall allow the [authorized administrator] to specify alternative initial values to override the default values when an object or information is created.*

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

5.1.5.16.1 FMT_MSA.3.1(e)

*The **Web Server part of the TSF** shall enforce the [CONTENT-PROVIDER SFP] to provide [restrictive] default values for security attributes that are used to enforce the SFP.*

5.1.5.16.2 FMT_MSA.3.2(e)

*The **Web Server part of the TSF** shall allow the [authorized administrator] to specify alternative initial values to override the default values when an object or information is created.*

5.1.5.17 Management of the Audit Trail (FMT_MTD.1(a))

5.1.5.17.1 FMT_MTD.1.1(a)

The TSF shall restrict the ability to create, delete, and clear the audit trail to authorized administrators.

5.1.5.18 Management of Audited Events (FMT_MTD.1(b))

5.1.5.18.1 FMT_MTD.1.1(b)

The TSF shall restrict the ability to modify or observe the set of audited events to authorized administrators.

5.1.5.19 Management of User Attributes (FMT_MTD.1(c))

5.1.5.19.1 FMT_MTD.1.1(c)

The TSF shall restrict the ability to initialize and modify the user security attributes, other than authentication data [and private/public keys] to authorized administrators.

5.1.5.20 Management of Authentication Data (FMT_MTD.1(d))

5.1.5.20.1 FMT_MTD.1.1(d)

The TSF shall restrict the ability to initialize the authentication data to authorized administrators.

5.1.5.20.2 FMT_MTD.1.2(d)

The TSF shall restrict the ability to modify the authentication data to the following:

- a) authorized administrators; and
- b) users authorized to modify their own authentication data.

5.1.5.21 Management of Account Lock Out Duration (FMT_MTD.1(e))

5.1.5.21.1 FMT_MTD.1.1(e)

The TSF shall restrict the ability to [modify] the [duration the user account is disabled after the unsuccessful authentication attempts threshold is exceeded] to [authorized administrators].

5.1.5.22 Management of Minimum Password Length (FMT_MTD.1(f))

5.1.5.22.1 FMT_MTD.1.1(f)

The TSF shall restrict the ability to [modify] the [minimum allowable password length] to [authorized administrators].

5.1.5.23 Management of TSF Time (FMT_MTD.1(g))

5.1.5.23.1 FMT_MTD.1.1(g)

The TSF shall restrict the ability to [modify] the [TSF representation of time] to [authorized administrators].

5.1.5.24 Management of NTFS Volume Quota Settings (FMT_MTD.1(h))**5.1.5.24.1 FMT_MTD.1.1(h)**

The TSF shall restrict the ability to [modify] the [quota settings on NTFS volumes] to [authorized administrators].

5.1.5.25 Management of Advisory Warning Message (FMT_MTD.1(i))**5.1.5.25.1 FMT_MTD.1.1(i)**

The TSF shall restrict the ability to [modify] the [advisory warning message displayed before establishing a user session] to [authorized administrators].

5.1.5.26 Management Audit Log Size (FMT_MTD.1(j))**5.1.5.26.1 FMT_MTD.1.1(j)**

The TSF shall restrict the ability to [modify] the [the audit log size] to [authorized administrators].

5.1.5.27 Management of User Inactivity Threshold (FMT_MTD.1(k))**5.1.5.27.1 FMT_MTD.1.1(k)**

The TSF shall restrict the ability to [change default, modify, delete, clear] the [user inactivity threshold for an authorized user during an interactive session] to [the authorized user].

5.1.5.28 Management of TSF Data (for general TSF data) (FMT_MTD.1(l))**5.1.5.28.1 FMT_MTD.1.1(l)**

The TSF shall restrict the ability to [create, change_default, query, modify, delete, and clear] the [security-relevant TSF data except for audit records, user security attributes, authentication data, and critical cryptographic security parameters] to [the authorized administrator.]

5.1.5.29 Management of TSF Data (for reading of authentication data) (FMT_MTD.1(m))**5.1.5.29.1 FMT_MTD.1.1(m)**

*The TSF shall **prevent** the ~~restrict the ability to~~ [reading] of [authentication data]. ~~to [the authorized identified roles].~~*

5.1.5.30 Management of Password Complexity Requirement (FMT_MTD.1(n))**5.1.5.30.1 FMT_MTD.1.1(n)**

The TSF shall restrict the ability to [modify] the [password complexity requirement] to [authorized administrators].

5.1.5.31 Management of User Private/Public Key Pair (FMT_MTD.1(o))**5.1.5.31.1 FMT_MTD.1.1(o)**

The TSF shall restrict the ability to [initialize] the [user security attributes private/public key pair] to [authorized administrators and authorized users].

5.1.5.32 Management of Unsuccessful Authentication Attempts Threshold (FMT_MTD.2)

5.1.5.32.1 FMT_MTD.2.1

The TSF shall restrict the specification of the limits for [the unsuccessful authentication attempts threshold] to [authorized administrators].

5.1.5.32.2 FMT_MTD.2.2

The TSF shall take the following action, if the TSF data are at, or exceed, the indicated limits: [the TSF shall disable the user account for an authorized administrator specified duration].

5.1.5.33 Revocation of User Attributes (FMT_REV.1(a))

5.1.5.33.1 FMT_REV.1.1(a)

The TSF shall restrict the ability to revoke security attributes associated with the users within the TSC to authorized administrators.

5.1.5.33.2 FMT_REV.1.2(a)

The TSF shall enforce the rules:

- a) The immediate revocation of security-relevant authorizations; and
- b) [No additional rule].

5.1.5.34 Revocation of Object Attributes (FMT_REV.1(b))

5.1.5.34.1 FMT_REV.1.1(b)

The TSF shall restrict the ability to revoke security attributes associated with **named objects** within the TSC to users authorized to modify the security attributes by the Discretionary Access Control policy.

5.1.5.34.2 FMT_REV.1.2(b)

The TSF shall enforce the rules:

- a) The access rights associated with an object shall be enforced when an access check is made; and
- b) [No additional rule].

5.1.5.35 Time-limited Authorization (FMT_SAE.1)

5.1.5.35.1 FMT_SAE.1.1

The TSF shall restrict the capability to specify an expiration time for [authentication data] to [authorized administrators].

5.1.5.35.2 FMT_SAE.1.2

For each of these security attributes, the TSF shall be able to [lock out the associated user account] after the expiration time for the indicated security attribute has passed.

5.1.5.36 Specification of Management Functions (FMT_SMF.1)

5.1.5.36.1 FMT_SMF.1.1

The TSF shall be capable of performing the following security management functions:

- [
- a) modify access control attributes associated with an object
 - b) delete encryption policy attributes associated with a file

- c) enable, disable, modify the behaviour of the audit function
 - d) determine and modify the behaviour of the function that protects data during transmission between parts of the TOE
 - e) modify the behaviour of the locked user session function
 - f) clear the audit trail
 - g) modify the set of events to be audited
 - h) read the audited events
 - i) initialize and modify user security attributes
 - j) modify the duration the user account is disabled after the unsuccessful authentication attempts threshold is exceeded
 - k) modify the minimum allowable password length
 - l) modify the quota settings on NTFS volumes
 - m) modify the advisory warning message displayed before establishment of a user session
 - n) modify the audit log size
 - o) modify the password complexity restriction
 - p) modify the unsuccessful authentication attempts threshold
 - q) modify the time
-].

5.1.5.37 Security Roles (FMT_SMR.1)

5.1.5.37.1 FMT_SMR.1.1

The TSF shall maintain the roles:

- a) Authorized administrator;
- b) Users authorized by the Discretionary Access Control Policy to modify object security attributes;
- c) Users authorized to modify their own authentication data and unlock the local user session; and
- d) [object creator - Users that create objects].

5.1.5.37.2 FMT_SMR.1.2

The TSF shall be able to associate users with roles.

5.1.5.38 Assuming Roles (FMT_SMR.3)

5.1.5.38.1 FMT_SMR.3.1

The TSF shall require an explicit request to assume ~~the following roles: [assignment: the roles]~~ **any role**.

5.1.6 Protection of the TOE Security Functions (FPT)

5.1.6.1 Internal Data Transfer Protection (TRANSFER_PROT_EX.1)

5.1.6.1.1 TRANSFER_PROT_EX.1.1

The TSF shall be able to protect data from disclosure and modification when it is transmitted between separate parts of the TOE through the use of encryption.

5.1.6.2 Internal TSF Data Integrity Monitoring (TRANSFER_PROT_EX.3)

5.1.6.2.1 TRANSFER_PROT_EX.3.1

The TSF shall be able to detect [modification, insertion and replay of data] for data transmitted between separate parts of the TOE through the use of cryptographic means.

5.1.6.2.2 TRANSFER_PROT_EX.3.2

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

- [
- a) reject data
- b) audit event
-]

5.1.6.3 Abstract Machine Testing (FPT_AMT.1)

5.1.6.3.1 FPT_AMT.1

The TSF shall run a suite of tests **during Windows Server 2003 Common Criteria evaluation** to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF

5.1.6.4 Replay Detection (FPT_RPL_EX.1)

5.1.6.4.1 FPT_RPL_EX.1.1

The TSF shall be able to detect replay of TSF data transmitted between separate parts of the TOE through the use of cryptographic means.

5.1.6.5 Internal TSF Data Consistency (FPT_TRC_EX.1)

5.1.6.5.1 FPT_TRC_EX.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 upon replication between parts of the TOE.

5.1.6.6 Non-bypassability of the TSP (FPT_RVM.1)

5.1.6.6.1 FPT_RVM.1.1

The TSF shall ensure that the TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.1.6.7 TSF Domain Separation (FPT_SEP.1)

5.1.6.7.1 FPT_SEP.1.1

The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

5.1.6.7.2 FPT_SEP.1.2

The TSF shall enforce separation between the security domains of subjects in the TSC.

5.1.6.8 Reliable Time Stamp (FPT_STM.1)

5.1.6.8.1 FPT_STM.1.1

The TSF shall be able to provide reliable time stamps for its own use.

5.1.7 Resource Utilization (FRU)

5.1.7.1 Maximum Quotas (FRU_RSA.1)

5.1.7.1.1 FRU_RSA.1.1

The TSF shall enforce maximum quotas of the following resources: [NTFS volumes] that [individual users] can use [simultaneously].

5.1.8 TOE Access (FTA)

5.1.8.1 Limitation on Scope of Selectable Attributes (FTA_LSA_EX.1)

5.1.8.1.1 FTA_LSA_EX.1.1

The TSF shall restrict the scope of session security attributes [roles and user privileges], based on [location, time, and day] if part of a domain.

5.1.8.2 Basic Limitation on Multiple Concurrent Sessions (FTA_MCS_EX.1)

5.1.8.2.1 FTA_MCS_EX.1.1

The TSF shall enforce a maximum number of concurrent interactive sessions per user, if part of a domain.

5.1.8.2.2 FTA_MCS_EX.1.2

The TSF shall allow an authorized administrator to set the maximum number of concurrent interactive sessions per user, if part of a domain.

5.1.8.3 TSF-Initiated Session Locking (FTA_SSL.1)

5.1.8.3.1 FTA_SSL.1.1

The TSF shall lock an interactive session after [a user-selected interval of inactivity or an 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.

5.1.8.3.2 FTA_SSL.1.2

The TSF shall require the following events to occur prior to unlocking the session:
[Re-authenticate the user.]

5.1.8.4 User-Initiated Session Locking (FTA_SSL.2)

5.1.8.4.1 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.

5.1.8.4.2 FTA_SSL.2.2

The TSF shall require the following events to occur prior to unlocking the session:
[Re-authenticate the user.]

5.1.8.5 WEBUSER TSF-Initiated Termination (FTA_SSL.3)

5.1.8.5.1 FTA_SSL.3.1

The **Web Server part of the** TSF shall terminate a **remote** interactive **http:// or https://** session after [an administrator configurable time interval of session inactivity].

5.1.8.6 Default TOE Access Banners (FTA_TAB.1)

5.1.8.6.1 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.

5.1.8.7 TOE Session Establishment (FTA_TSE.1)

5.1.8.7.1 FTA_TSE.1.1

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

5.1.9 Trusted Path/Channels

5.1.9.1 Trusted Path (FTP_TRP.1)

5.1.9.1.1 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.

5.1.9.1.2 FTP_TRP.1.2

The TSF shall permit [local and remote users] to initiate the communication via the trusted path.

5.1.9.1.3 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, session unlocking, and changing user password when the TSF requests/notifications (via the trusted path) the user of the user account, to which the password belongs, to change password].

5.2 TOE SARs

The SARs for the TOE are the EAL 4 components augmented with ALC_FLR.3 as specified in Part 3 of the CC. No operations are applied to the assurance components.

Interpreted Requirements

Requirements that have been modified based upon an International Interpretation are identified by an italicized parenthetical comment following the requirement element that has been modified (e.g. *(per International Interpretation #51)*).

Table 5-5 EAL 4 Assurance Components

Assurance Class	Assurance Components
Configuration Management (ACM)	ACM_AUT.1 Partial CM Automation
	ACM_CAP.4 Generation Support and Acceptance Procedures
	ACM_SCP.2 Problem Tracking CM Coverage
Delivery and Operation (ADO)	ADO_DEL.2 Detection of Modification
	ADO_IGS.1 Installation, Generation, and Start-up Procedures
Development (ADV)	ADV_FSP.2 Fully Defined External Interfaces
	ADV_HLD.2 Security Enforcing High-level Design
	ADV_IMP.1 Subset of the Implementation of the TSF
	ADV_LLD.1 Descriptive Low-level Design
	ADV_RCR.1 Informal Correspondence Demonstration
	ADV_SPM.1 Informal TOE Security Policy Model
Guidance Documents (AGD)	AGD_ADM.1 Administrator Guidance
	AGD_USR.1 User Guidance
Life cycle support (ALC)	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
Tests (ATE)	ATE_COV.2 Analysis of Coverage
	ATE_DPT.1 Testing: High-level Design
	ATE_FUN.1 Functional Testing
	ATE_IND.2 Independent Testing – Sample
Vulnerability assessment (AVA)	AVA_MSU.2 Validation of Analysis
	AVA_SOF.1 Strength of TOE Security Function Evaluation
	AVA_VLA.2 Independent Vulnerability Analysis

5.2.1 Configuration Management (ACM)

5.2.1.1 Partial CM Automation (ACM_AUT.1)

5.2.1.1.1 ACM_AUT.1.1D

The developer shall use a CM system.

5.2.1.1.2 ACM_AUT.1.2D

The developer shall provide a CM plan.

5.2.1.1.3 ACM_AUT.1.1C

The CM system shall provide an automated means by which only authorized changes are made to the TOE implementation representation.

5.2.1.1.4 ACM_AUT.1.2C

The CM system shall provide an automated means to support the generation of the TOE.

5.2.1.1.5 ACM_AUT.1.3C

The CM plan shall describe the automated tools used in the CM system.

5.2.1.1.6 ACM_AUT.1.4C

The CM plan shall describe how the automated tools are used in the CM system.

5.2.1.1.7 ACM_AUT.1.1E

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

5.2.1.2 Generation Support and Acceptance Procedures (ACM_CAP.4)

5.2.1.2.1 ACM_CAP.4.1D

The developer shall provide a reference for the TOE.

5.2.1.2.2 ACM_CAP.4.2D

The developer shall use a CM system.

5.2.1.2.3 ACM_CAP.4.3D

The developer shall provide CM documentation.

5.2.1.2.4 ACM_CAP.4.1C

The reference for the TOE shall be unique to each version of the TOE.

5.2.1.2.5 ACM_CAP.4.2C

The TOE shall be labeled with its reference.

5.2.1.2.6 ACM_CAP.4.3C

The CM documentation shall include a configuration list, a CM plan, and an acceptance plan.

5.2.1.2.7 ACM_CAP.4.NewC

The configuration list shall uniquely identify all configuration items that comprise the TOE. (*per International Interpretation #3*)

5.2.1.2.8 ACM_CAP.4.4C

The configuration list shall describe the configuration items that comprise the TOE.

5.2.1.2.9 ACM_CAP.4.5C

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

5.2.1.2.10 ACM_CAP.4.6C

The CM system shall uniquely identify all configuration items.

5.2.1.2.11 ACM_CAP.4.7C

The CM plan shall describe how the CM system is used.

5.2.1.2.12 ACM_CAP.4.8C

The evidence shall demonstrate that the CM system is operating in accordance with the CM plan.

5.2.1.2.13 ACM_CAP.4.9C

The CM documentation shall provide evidence that all configuration items have been and are being effectively maintained under the CM system.

5.2.1.2.14 ACM_CAP.4.10C

The CM system shall provide measures such that only authorized changes are made to the configuration items.

5.2.1.2.15 ACM_CAP.4.11C

The CM system shall support the generation of the TOE.

5.2.1.2.16 ACM_CAP.4.12C

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

5.2.1.2.17 ACM_CAP.4.1E

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

5.2.1.3 Problem Tracking CM Coverage (ACM_SCP.2)

5.2.1.3.1 ACM_SCP.2.1D

The developer shall provide a list of configuration items for the TOE. (*per International Interpretation #4*).

5.2.1.3.2 ACM_SCP.2.1C

The list of configuration items shall include the following: implementation representation; security flaws; and the evaluation evidence required by the assurance components in the ST. (*per International Interpretation #4 and #38*).

5.2.1.3.3 ACM_SCP.2.2C

(this element has been deleted per International Interpretation #4)

5.2.1.3.4 ACM_SCP.2.1E

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

5.2.2 Delivery and Operation (ADO)

5.2.2.1 Detection of Modification (ADO_DEL.2)

5.2.2.1.1 ADO_DEL.2.1D

The developer shall document procedures for delivery of the TOE or parts of it to the user.

5.2.2.1.2 ADO_DEL.2.2D

The developer shall use the delivery procedures.

5.2.2.1.3 ADO_DEL.2.1C

The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

5.2.2.1.4 ADO_DEL.2.2C

The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.

5.2.2.1.5 ADO_DEL.2.3C

The delivery documentation shall describe how the various procedures allow detection of attempts to masquerade as the developer, even in cases in which the developer has sent nothing to the user's site.

5.2.2.1.6 ADO_DEL.2.1E

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

5.2.2.2 Installation, Generation, and Start-up Procedures (ADO_IGS.1)

5.2.2.2.1 ADO_IGS.1.1D

The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

5.2.2.2.2 ADO_IGS.1.1C

The installation, generation and start-up documentation shall describe all the steps necessary for secure installation, generation and start-up of the TOE. (*per International Interpretation # 51*)

5.2.2.2.3 ADO_IGS.1.1E

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

5.2.2.2.4 ADO_IGS.1.2E

The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.2.3 Development (ADV)

5.2.3.1 Fully Defined External Interfaces (ADV_FSP.2)

5.2.3.1.1 ADV_FSP.2.1D

The developer shall provide a functional specification.

5.2.3.1.2 ADV_FSP.2.1C

The functional specification shall describe the TSF and its external interfaces using an informal style.

5.2.3.1.3 ADV_FSP.2.2C

The functional specification shall be internally consistent.

5.2.3.1.4 ADV_FSP.2.3C

The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing complete details of all effects, exceptions and error messages.

5.2.3.1.5 ADV_FSP.2.4C

The functional specification shall completely represent the TSF.

5.2.3.1.6 ADV_FSP.2.5C

The functional specification shall include rationale that the TSF is completely represented.

5.2.3.1.7 ADV_FSP.2.1E

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

5.2.3.1.8 ADV_FSP.2.2E

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

5.2.3.2 Security Enforcing High-level Design (ADV_HLD.2)

5.2.3.2.1 ADV_HLD.2.1D

The developer shall provide the high-level design of the TSF.

5.2.3.2.2 ADV_HLD.2.1C

The presentation of the high-level design shall be informal.

5.2.3.2.3 ADV_HLD.2.2C

The high-level design shall be internally consistent.

5.2.3.2.4 ADV_HLD.2.3C

The high-level design shall describe the structure of the TSF in terms of subsystems.

5.2.3.2.5 ADV_HLD.2.4C

The high-level design shall describe the security functionality provided by each subsystem of the TSF.

5.2.3.2.6 ADV_HLD.2.5C

The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.

5.2.3.2.7 ADV_HLD.2.6C

The high-level design shall identify all interfaces to the subsystems of the TSF.

5.2.3.2.8 ADV_HLD.2.7C

The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.

5.2.3.2.9 ADV_HLD.2.8C

The high-level design shall describe the purpose and method of use of all interfaces to the subsystems of the TSF, providing details of effects, exceptions and error messages, as appropriate.

5.2.3.2.10 ADV_HLD.2.9C

The high-level design shall describe the separation of the TOE into TSP-enforcing and other subsystems.

5.2.3.2.11 ADV_HLD.2.1E

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

5.2.3.2.12 ADV_HLD.2.2E

The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.2.3.3 **Subset of the Implementation of the TSF (ADV_IMP.1)**

5.2.3.3.1 ADV_IMP.1.1D

The developer shall provide the implementation representation for a selected subset of the TSF.

5.2.3.3.2 ADV_IMP.1.1C

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

5.2.3.3.3 ADV_IMP.1.2C

The implementation representation shall be internally consistent.

5.2.3.3.4 ADV_IMP.1.1E

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

5.2.3.3.5 ADV_IMP.1.2E

The evaluator shall determine that the least abstract TSF representation provided is an accurate and complete instantiation of the TOE security functional requirements.

5.2.3.4 Descriptive Low-level Design (ADV_LLD.1)

5.2.3.4.1 ADV_LLD.1.1D

The developer shall provide the low-level design of the TSF.

5.2.3.4.2 ADV_LLD.1.1C

The presentation of the low-level design shall be informal.

5.2.3.4.3 ADV_LLD.1.2C

The low-level design shall be internally consistent.

5.2.3.4.4 ADV_LLD.1.3C

The low-level design shall describe the TSF in terms of modules.

5.2.3.4.5 ADV_LLD.1.4C

The low-level design shall describe the purpose of each module.

5.2.3.4.6 ADV_LLD.1.5C

The low-level design shall define the interrelationships between the modules in terms of provided security functionality and dependencies on other modules.

5.2.3.4.7 ADV_LLD.1.6C

The low-level design shall describe how each TSP-enforcing function is provided.

5.2.3.4.8 ADV_LLD.1.7C

The low-level design shall identify all interfaces to the modules of the TSF.

5.2.3.4.9 ADV_LLD.1.8C

The low-level design shall identify which of the interfaces to the modules of the TSF are externally visible.

5.2.3.4.10 ADV_LLD.1.9C

The low-level design shall describe the purpose and method of use of all interfaces to the modules of the TSF, providing details of effects, exceptions and error messages, as appropriate.

5.2.3.4.11 ADV_LLD.1.10C

The low-level design shall describe the separation of the TOE into TSP-enforcing and other modules.

5.2.3.4.12 ADV_LLD.1.1E

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

5.2.3.4.13 ADV_LLD.1.2E

The evaluator shall determine that the low-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.2.3.5 Informal Correspondence Demonstration (ADV_RCR.1)

5.2.3.5.1 ADV_RCR.1.1D

The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.

5.2.3.5.2 ADV_RCR.1.1C

For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.

5.2.3.5.3 ADV_RCR.1.1E

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

5.2.3.6 Informal TOE Security Policy Model (ADV_SPM.1)

5.2.3.6.1 ADV_SPM.1.1D

The developer shall provide a TSP model.

5.2.3.6.2 ADV_SPM.1.2D

The developer shall demonstrate correspondence between the functional specification and the TSP model.

5.2.3.6.3 ADV_SPM.1.1C

The TSP model shall be informal.

5.2.3.6.4 ADV_SPM.1.2C

The TSP model shall describe the rules and characteristics of all policies of the TSP that can be modeled.

5.2.3.6.5 ADV_SPM.1.3C

The TSP model shall include a rationale that demonstrates that it is consistent and complete with respect to all policies of the TSP that can be modeled.

5.2.3.6.6 ADV_SPM.1.4C

The demonstration of correspondence between the TSP model and the functional specification shall show that all of the security functions in the functional specification are consistent and complete with respect to the TSP model.

5.2.3.6.7 ADV_SPM.1.1E

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

5.2.4 Guidance Documents (AGD)

5.2.4.1 Administrator Guidance (AGD_ADM.1)

5.2.4.1.1 AGD_ADM.1.1D

The developer shall provide administrator guidance addressed to system administrative personnel.

5.2.4.1.2 AGD_ADM.1.1C

The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.

5.2.4.1.3 AGD_ADM.1.2C

The administrator guidance shall describe how to administer the TOE in a secure manner.

5.2.4.1.4 AGD_ADM.1.3C

The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.

5.2.4.1.5 AGD_ADM.1.4C

The administrator guidance shall describe all assumptions regarding user behavior that are relevant to secure operation of the TOE.

5.2.4.1.6 AGD_ADM.1.5C

The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.

5.2.4.1.7 AGD_ADM.1.6C

The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

5.2.4.1.8 AGD_ADM.1.7C

The administrator guidance shall be consistent with all other documentation supplied for evaluation.

5.2.4.1.9 AGD_ADM.1.8C

The administrator guidance shall describe all security requirements on the IT environment that are relevant to the administrator.

5.2.4.1.10 AGD_ADM.1.1E

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

5.2.4.2 User Guidance (AGD_USR.1)

5.2.4.2.1 AGD_USR.1.1D

The developer shall provide user guidance.

5.2.4.2.2 AGD_USR.1.1C

The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

5.2.4.2.3 AGD_USR.1.2C

The user guidance shall describe the use of user-accessible security functions provided by the TOE.

5.2.4.2.4 AGD_USR.1.3C

The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.

5.2.4.2.5 AGD_USR.1.4C

The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behavior found in the statement of TOE security environment.

5.2.4.2.6 AGD_USR.1.5C

The user guidance shall be consistent with all other documentation supplied for evaluation.

5.2.4.2.7 AGD_USR.1.6C

The user guidance shall describe all security requirements on the IT environment that are relevant to the user.

5.2.4.2.8 AGD_USR.1.1E

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

5.2.5 Life Cycle Support (ALC)

5.2.5.1 Identification of Security Measures (ALC_DVS.1)

5.2.5.1.1 ALC_DVS.1.1D

The developer shall produce development security documentation.

5.2.5.1.2 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.

5.2.5.1.3 ALC_DVS.1.2C

The development security documentation shall provide evidence that these security measures are followed during the development and maintenance of the TOE.

5.2.5.1.4 ALC_DVS.1.1E

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

5.2.5.1.5 ALC_DVS.1.2E

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

5.2.5.2 Systematic Flaw Remediation (ALC_FLR.3)

5.2.5.2.1 ALC_FLR.3.1D

The developer shall document the flaw remediation procedures.

5.2.5.2.2 ALC_FLR.3.2D

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

5.2.5.2.3 ALC_FLR.3.3D

The developer shall designate one or more specific points of contact for user reports and inquiries about security issues involving the TOE.

5.2.5.2.4 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.

5.2.5.2.5 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.

5.2.5.2.6 ALC_FLR.3.3C

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

5.2.5.2.7 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.

5.2.5.2.8 ALC_FLR.3.5C

The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to TOE users.

5.2.5.2.9 ALC_FLR.3.6C

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

5.2.5.2.10 ALC_FLR.3.7C

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

5.2.5.2.11 ALC_FLR.3.1E

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

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

5.2.5.3.1 ALC_LCD.1.1D

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

5.2.5.3.2 ALC_LCD.1.2D

The developer shall provide life-cycle definition documentation.

5.2.5.3.3 ALC_LCD.1.1C

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

5.2.5.3.4 ALC_LCD.1.2C

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

5.2.5.3.5 ALC_LCD.1.1E

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

5.2.5.4 Well-defined Development Tools (ALC_TAT.1)

5.2.5.4.1 ALC_TAT.1.1D

The developer shall identify the development tools being used for the TOE.

5.2.5.4.2 ALC_TAT.1.2D

The developer shall document the selected implementation-dependent options of the development tools.

5.2.5.4.3 ALC_TAT.1.1C

All development tools used for implementation shall be well defined.

5.2.5.4.4 ALC_TAT.1.2C

The documentation of the development tools shall unambiguously define the meaning of all statements used in the implementation.

5.2.5.4.5 ALC_TAT.1.3C

The documentation of the development tools shall unambiguously define the meaning of all implementation-dependent options.

5.2.5.4.6 ALC_TAT.1.1E

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

5.2.6 Security Testing (ATE)

5.2.6.1 Analysis of Coverage (ATE_COV.2)

5.2.6.1.1 ATE_COV.2.1D

The developer shall provide an analysis of the test coverage.

5.2.6.1.2 ATE_COV.2.1C

The analysis of the test coverage shall demonstrate the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

5.2.6.1.3 ATE_COV.2.2C

The analysis of the test coverage shall demonstrate that the correspondence between the TSF as described in the functional specification and the tests identified in the test documentation is complete.

5.2.6.1.4 ATE_COV.2.1E

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

5.2.6.2 Testing: High-level Design (ATE_DPT.1)

5.2.6.2.1 ATE_DPT.1.1D

The developer shall provide the analysis of the depth of testing.

5.2.6.2.2 ATE_DPT.1.1C

The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the TSF operates in accordance with its high-level design.

5.2.6.2.3 ATE_DPT.1.1E⁹

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

5.2.6.3 Functional Testing (ATE_FUN.1)

5.2.6.3.1 ATE_FUN.1.1D

The developer shall test the TSF and document the results.

5.2.6.3.2 ATE_FUN.1.2D

The developer shall provide test documentation.

5.2.6.3.3 ATE_FUN.1.1C

The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

5.2.6.3.4 ATE_FUN.1.2C

The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.

5.2.6.3.5 ATE_FUN.1.3C

The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.

5.2.6.3.6 ATE_FUN.1.4C

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

5.2.6.3.7 ATE_FUN.1.5C

The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.

⁹ This label is consistent with the CAPP. In the CC, this element is incorrectly labeled as “ATE_DPT.1.2E”

5.2.6.3.8 ATE_FUN.1.1E

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

5.2.6.4 Independent Testing – Sample (ATE_IND.2)

5.2.6.4.1 ATE_IND.2.1D

The developer shall provide the TOE for testing.

5.2.6.4.2 ATE_IND.2.1C

The TOE shall be suitable for testing.

5.2.6.4.3 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.

5.2.6.4.4 ATE_IND.2.1E

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

5.2.6.4.5 ATE_IND.2.2E

The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.

5.2.6.4.6 ATE_IND.2.3E

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

5.2.7 Vulnerability Assessment (AVA)

5.2.7.1 Validation of Analysis (AVA_MSU.2)

5.2.7.1.1 AVA_MSU.2.1D

The developer shall provide guidance documentation.

5.2.7.1.2 AVA_MSU.2.2D

The developer shall document an analysis of the guidance documentation.

5.2.7.1.3 AVA_MSU.2.1C

The guidance documentation 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.

5.2.7.1.4 AVA_MSU.2.2C

The guidance documentation shall be complete, clear, consistent and reasonable.

5.2.7.1.5 AVA_MSU.2.3C

The guidance documentation shall list all assumptions about the intended environment.

5.2.7.1.6 AVA_MSU.2.4C

The guidance documentation shall list all requirements for external security measures (including external procedural, physical and personnel controls).

5.2.7.1.7 AVA_MSU.2.5C

The analysis documentation shall demonstrate that the guidance documentation is complete.

5.2.7.1.8 AVA_MSU.2.1E

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

5.2.7.1.9 AVA_MSU.2.2E

The evaluator shall repeat all configuration and installation procedures, and other procedures selectively, to confirm that the TOE can be configured and used securely using only the supplied guidance documentation.

5.2.7.1.10 AVA_MSU.2.3E

The evaluator shall determine that the use of the guidance documentation allows all insecure states to be detected.

5.2.7.1.11 AVA_MSU.2.4E

The evaluator shall confirm that the analysis documentation shows that guidance is provided for secure operation in all modes of operation of the TOE.

5.2.7.2 Strength of TOE Security Function Evaluation (AVA_SOF.1)

5.2.7.2.1 AVA_SOF.1.1D

The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.

5.2.7.2.2 AVA_SOF.1.1C

For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.

5.2.7.2.3 AVA_SOF.1.2C

For each mechanism with a specific strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the specific strength of function metric defined in the PP/ST.

5.2.7.2.4 AVA_SOF.1.1E

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

5.2.7.2.5 AVA_SOF.1.2E

The evaluator shall confirm that the strength claims are correct.

5.2.7.3 Independent Vulnerability Analysis (AVA_VLA.2)

5.2.7.3.1 AVA_VLA.2.1D

The developer shall perform a vulnerability analysis.

5.2.7.3.2 AVA_VLA.2.2D

The developer shall provide vulnerability analysis documentation.

5.2.7.3.3 AVA_VLA.2.1C

The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for ways in which a user can violate the TSP.

5.2.7.3.4 AVA_VLA.2.2C

The vulnerability analysis documentation shall describe the disposition of identified vulnerabilities.

5.2.7.3.5 AVA_VLA.2.3C

The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.

5.2.7.3.6 AVA_VLA.2.4C

The vulnerability analysis documentation shall justify that the TOE, with the identified vulnerabilities, is resistant to obvious penetration attacks.

5.2.7.3.7 AVA_VLA.2.1E

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

5.2.7.3.8 AVA_VLA.2.2E

The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure the identified vulnerabilities have been addressed.

5.2.7.3.9 AVA_VLA.2.3E

The evaluator shall perform an independent vulnerability analysis.

5.2.7.3.10 AVA_VLA.2.4E

The evaluator shall perform independent penetration testing, based on the independent vulnerability analysis, to determine the exploitability of additional identified vulnerabilities in the intended environment.

5.2.7.3.11 AVA_VLA.2.5E

The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a high attack potential.

5.3 Security Requirements for the IT Environment

The TOE has no security requirements allocated to its IT environment.

6. TOE Summary Specification (TSS)

This chapter describes the Windows 2003/XP security functions and associated assurance measures. The Windows 2003/XP Security Functions (SFs) and Security Assurance Measures (SAMs) satisfy the security functional and assurance requirements of the CAPP. The TOE also satisfies additional SFs and SAMs. The SFs and SAMs performed by Windows 2003/XP are described in the following sections, as well as a mapping to the security functional and assurance requirement satisfied by the TOE.

6.1 TOE Security Functions

This section presents the TSFs and a mapping of security functions to SFRs. The TOE performs the following security functions:

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

6.1.1 Audit Function

The TOE Audit security function performs:

- Audit Collection,
- Audit Log Review,
- Selective Audit,
- Audit Log Overflow Protection, and
- 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 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:	This will be "Security."

- Types: Indicates whether the security audit event recorded is the result of a successful or failed attempt to perform the action.
- Category: A classification of the event defined by the event source. For 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 such as described below:

- For the System Category, the audit records additionally include information relating to the system such as the time of clearing the audit trail.
- 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 by indicating 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 request 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 with the LSA server or that 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, allowing a more precise definition of the audit policy.

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 Tables 5-3 and 5-4. For each category, the associated audit events (listed in Tables 5-3 and 5-4) 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; FAU_STG.4; FMT_MTD.1(a)
Object Access	Audit attempts to access user objects, such as files.	FDP_ACF.1(a); FMT_MSA.1(a); FMT_MSA.3(a); FMT_REV(b);
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.	FMT_SMR.1; FPT_STM.1; FMT_MTD.1(g); FMT_MOF.1(a); FMT_MTD.1(a); FAU_SAR.1; FAU_SAR.2
Detailed Process Tracking	Audit subject-tracking events, including program activation, handle duplication, indirect access to an object, and process exit.	FIA_USB.1_EX; FDP_ACF.1(a); FMT_MSA.1(d)
Policy Change	Audit attempts to change security policy settings such as the audit policy and privilege assignment.	FMT_MTD.1(b); FMT_MTD.1(c); FMT_REV.1(a); FMT_SMR.1; FMT_MOF.1(a); TRANSFER_PROT_EX.1; TRANSFER_PROT_EX.3; FAU_GEN.1
Account Management	Audit attempts to create, delete, or change user or group accounts and changes to their attributes.	FMT_MTD.1(c); FMT_MTD.1(d); FMT_REV.1(a); FMT_SMR.1; FIA_AFL.1; FMT_SAE.1; FMT_MTD.1(f); FMT_MTD.1(n); FMT_MTD.2; FMT_MTD.1(e)
Directory Service Access	Audit access to directory service objects and associated properties.	FDP_ACF.1(a) ; FPT_TRC_EX.1

Category	Description	FAU_GEN Required Events
Logon	Audit attempts to logon or logoff the system, attempts to make a network connection.	FIA_SOS.1; FIA_UAU.1; FIA_UID.1; FIA_AFL.1; FIA_USB.1; FTA_SSL.1; FTA_SSL.2; FTA_TSE.1; TRANSFER_PROT_EX; FTP_TRP.1
Account Logon	Audit when a DC receives a logon request.	FIA_SOS.1; FIA_UAU.1; FIA_UID.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, date, time, source, category, event ID, and computer. The security log can also be searched by free form texts occurring in the audit records.

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), event type, and 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 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. 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.1(a) – Only authorized administrators have any access to the audit log.
 - FAU_SAR.3(a), (b) – 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.
 - 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.1(j) – 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,
- WEBUSER Access Control,
- Content Provider Access Control,
- Information Flow Control and Protection, and
- Residual Data Protection.

6.1.2.1 Discretionary Access Control

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. The interactive window station has three default desktops created by WinLogon.
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 Mutex at the win32 interface).
Object Directory	A directory in the object namespace.
LPC Port	A connection-oriented local process communication mechanism object that supports client and server side communication end points, message queues, etc.
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.
Process	An execution context for threads that has associated address space and memory, token, handle table, etc.
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, etc.) 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.
Tokens	These objects represent 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.

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 SID for the user, SIDs representing groups for which the user is a member, privileges assigned to the user, an owner SID identifying SID to assign as owner for newly created objects, a default DACL (for newly created objects), token type (primary or impersonation), 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).

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

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:

- 1 ALLOW ACES
 - a. ACCESS_ALLOWED_ACE – used to grant access to a user or group of users
 - b. 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.
- 2 DENY ACES
 - a. ACCESS_DENIED_ACE – used to deny access to a user or group of users
 - b. 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.

An access mask contains object access rights granted (or denied) to the SID, representing a user or group, in the ACE. 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 that 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.
2. Owner Check –
 - a. Checks all SIDs in 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 associate 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 Types — If the ACE is of type `ACCESS_DENIED_OBJECT_ACE` and the ACE includes a GUID representing a property set or property associate 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.

6.1.2.1.4 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 FEK. The FEK is used to encrypt the file data using the Triple Data Encryption Standard (3DES) CBC algorithm. The TSF stores the FEK as an attribute of the file and encrypts the FEK using the RSA public-key based encryption algorithm with the user's public key. The TSF also allows a user who can decrypt the file to grant access to other users by adding additional encrypted FEKs (encrypted with the new 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 private key associated with the recovery agent is referred to as recovery keys. The TSF also encrypts the FEK with one or more recovery keys. 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.

Once 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.

6.1.2.1.5 Default DAC Protection

The TSF provides a process ensuring a DACL is applied to all new objects. When new objects are created, the appropriate DACL is determined. The default DAC protection for DS 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.
- 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 checks the subject's access token for a default DACL. 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 can provide 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 checks the subject's access token for a default DACL. 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.

6.1.2.2 WEBUSER Access Control

The TOE includes a web server (the IIS) on the Windows 2003 server product that mediates access request to its web server content from clients accessing the web server through the HTTP.

IIS supports user authentication using either anonymous, basic, digest, certificate, NT or passport 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 web content access. 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 configuration 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. These settings allow for access control checks to be performed by IIS when access request are made to these URLs, if configured. 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,
- Source web permission: access the source of web content, and
- Browsing web permission: view the lists and collections in a directory.

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

6.1.2.2.1 WEBUSER Data Integrity and Confidentiality

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 instead of HTTP 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.

6.1.2.3 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.

6.1.2.4 Information Flow Control and Protection

The TOE includes a homogenous set of Windows 2003/XP systems that can be connected via their network interfaces. Each Windows 2003/XP system within the TOE provides a subset of the TSFs. Therefore, the TSF for Windows 2003/XP 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 2003/XP 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. See Section 6.1.6.2, Internal TOE Protection, for further details of IPSec. Note that IPSEC and ISAKMP are not available when the TOE is configured to use IPv6 for sending or receiving network traffic.

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 flows such as encrypt or sign. IPSEC uses the Encapsulating Security Protocol (ESP) to provide data confidentiality for IP packets. ESP performs encryption using the Data Encryption Standard (DES) Cipher Block Chaining (CBC) algorithm and 3DES CBC. In addition to ESP, the TSF implements IP Authentication Header (AH). AH provides integrity, authentication and anti-replay. AH uses a hashing algorithm, such as Secure Hash Algorithm (SHA)-1, to compute a keyed message hash for each IP packet. See Section 6.1.5.2, 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. 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.

6.1.2.5 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.¹⁰ 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.2(a) – 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.

¹⁰ For APIs that create objects, the caller may provide data to initialize the object.

- FDP_ACF.1(a) – 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 the 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.2 (b), FDP_ACC.2(c), FDP_ACF.1(b), FDP_ACF.1(c) – 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.2(b), FDP_ACF.2(b)). 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.2(c), FDP_ACF.2(c)).
- FDP_IFC.1(a), FDP_IFF.1(a) – The TSF controls the flow of traffic from one Windows 2003/XP 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.1(b), FDP_IFF.1(b) – The TSF controls the flow of traffic into a Windows 2003/XP 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
- FMT_MOF.1(d) – Only an authorized administrator can modify the values in the metabase which include the IIS configuration. These values define permissions to web content.
- FMT.MSA.1(a) – 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.1(c) – The ability to change the security attributes upon which the IPSec Filter Policy is based upon is restricted to the authorized administrator.
- FMT_MSA.1(d) – The ability to change the security attributes upon which the Connection Firewall Policy is based upon is restricted to the authorized administrator.
- FMT_MSA.1(e), FMT_MSA1(f) – 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.3(a) - 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

DAACL to override the default. The initial keys are cryptographically generated and cannot be modified.

- FMT_MSA.3(b) – 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.3(c) – By default, the list of ports which the TSF will allow unsolicited traffic into a Windows 2003/XP system's TSF to is empty. 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.3(d), FMT_MSA.3(e) – 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_REV.1(b) – 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.1 above. The changed DACL is effective upon subsequent access checks against the object.
- FMT_MSA.1(b) – 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.
- 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.
- Note1_EX - Every process is allocated new memory and an execution context. Memory is zeroed or overwritten before allocation. The execution is initialized or restored when threads are created or when a context switch occurs.

6.1.3 Cryptographic Protection

The TOE CryptProtectData and CryptUnprotectData functions encrypt and decrypt data blocks preventing the data from being decrypted by unauthorized users. These functions are used internally by the TOE as well as by users. The TSF uses this function to encrypt the private keys used by the Rivest, Shamir and Adleman (RSA) public-key-based encryption algorithm which are used to decrypt the FEKs (FEK are described the DAC Enforcement of Encrypted Files). Users can use these functions to encrypt and decrypt raw data blocks at their discretion. Upon a request to encrypt data, the TSF creates a random, symmetric key, known as the user's master key to encrypt the user data. The integrity of the user's master key is ensured by the TSF based on SHA-1 Hash-Based Message Authentication Code (HMAC) which uses a hash of the user's master key, and a hash of the user's SID with the user's logon password (credential). A backup key of the user's master key is also calculated based on a backup/restore key of a DC, if present. The key size is based on the DES or 3DES algorithm used for encryption, and is 56 or 168 bits.

Using the user's master key, the TSF encrypts the data and the encrypted data is returned to the user. Upon a request to decrypt data provided by the user, the TSF uses the user's master key to decrypt the data. If the decryption fails, the user does not receive any data.

When discovering the user's master key, the TSF performs a Message Authentication Code (MAC) check to verify proper decryption of the user's master key. If the MAC check succeeds, the master key is used to decrypt the user's data. If the MAC check fails, then the TSF attempts to decrypt the data using the backup/restore master key. This may be necessary if the user's password has changed. The TSF verifies that the user is authorized to the backup/restore master key by comparing the SID extracted using the backup/restore master key to the SID of the caller. If the caller is authorized to the backup/restore master key (e.g., the SIDs match), then the backup/restore master key is used to decrypt the data. If this decryption fails also, the user does not receive any data.

The encryption and decryption operations are performed by independent modules, known as Cryptographic Service Providers (CSPs). The CSPs are FIPS 140 Level 1 compliant. The FIPS 140 standard was developed by the National Institute of Standards and Technology (NIST). FIPS 140, titled "Security Requirements for Cryptographic Modules," specifies the U.S. government's requirements for proper design and implementation of hardware and software cryptographic modules that perform cryptographic operations for sensitive but unclassified information. FIPS 140 has been adopted by the Canadian Communication Security Establishment (CCSE) and the American National Standards Institute (ANSI). FIPS 140 is widely regarded as a de facto standard for cryptographic modules.

In addition to encryption and decryption services, the TSF provides other cryptographic operations such as hashing, key exchange, and digital signatures. The TSF also provides a pseudo random number generation capability. 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 Method

Cryptographic Operation	Standard	Evaluation Method
Encryption/Decryption	FIPS 46-3 DES (ECB and CBC modes)	FIPS 140-2 Certified (382)
Encryption/Decryption	FIPS 46-3 3DES (ECB and CBC modes)	FIPS 140-2 Certified (382)
Encryption/Decryption	FIPS 197 (Advanced Encryption Standard (AES))	FIPS 140-2 Certified (382)
Encryption/Decryption	PKCS#1(RSAES-PKCS1-v1_5)	Vendor Assertion
HMAC	FIPS-198 HMAC (SHA-1)	FIPS 140-2 Certified (382)
Hash	FIPS-180-2 SHA-1, SHA-256 ¹¹ , SHA-384 ¹² , SHA-512 ¹³	FIPS 140-2 Certified (382)
Random number generation	None	Vendor Assertion
Key exchange	DH (Ephemeral-Ephemeral)	Vendor Assertion
Key Generation	Software RNG (FIPS 46-3 3DES and DES)	Vendor Assertion
Key Generation	Software RNG (FIPS 197 AES)	Vendor Assertion
Key Generation	Prime RNG for RSA	Vendor Assertion
Key Generation	RNG (FIPS 186-2 DSA)	Vendor Assertion
Key Generation	RNG (DH)	Vendor Assertion
Key Distribution	PKCS#5 V2.0 (Password-Based Encryption Standard)	Vendor Assertion
Key Zeroization	FIPS 140-1 or FIPS 140-2 ¹⁴	FIPS 140-2 Certified (382)

¹¹ SHA-256 is available on the 2003 Server product only.

¹² SHA-384 is available on the 2003 Server product only.

¹³ SHA-512 is available on the 2003 Server product only.

¹⁴ FIPS 140-1 and 140-2 certification includes specific key zeroization requirements.

Cryptographic Operation	Standard	Evaluation Method
Key Storage	PKCS#5 V2.0 (Password-Based Encryption Standard), using FIPS-180-2 SHA-1 and FIPS-46-3 Triple DES	Vendor Assertion

The TSF applies validation techniques (e.g., hash) to validate the CryptProtectData-encrypted FIPS-186-2 DSA and FIPS-186-2/PKCS#1 private asymmetric cryptographic keys when they are obtained via the distribution method or via the storage method. The TSF checks the hash of the encrypted data (which is meant to be a CryptProtectData-encrypted private key).

The TSF provides a means to ensure that FIPS-186-2 DSA and FIPS-186-2/PKCS#1 private asymmetric cryptographic keys are associated with the correct entities (i.e., person) to which the keys are assigned. The private keys are encrypted with a user masterkey. The TSF encrypts the user masterkey with a SHA-1 hash of the logon password of the key owner.

The TSF uses PKCS#5 V2.0 (Password-Based Encryption Standard), using FIPS-180-2 SHA-1 and FIPS-46-3 3DES in the process to store FIPS-186-2 DSA and FIPS-186-2/PKCS#1 private asymmetric cryptographic keys. The TSF encrypts the private keys with a master key using 3DES; and the masterkey is encrypted using the PKCS#5 format with a SHA-1 hashing of the key owner's interactively-logging-on password. The encrypted private keys and encrypted masterkeys are written to their persistent files. These files are stored in a subdirectory of the user profile so that they could be loaded when the profile owner logs onto a machine. The key storage effect is achieved because the encrypted private keys and encrypted masterkeys are stored in their persistent files in the user profile. User profiles can be centrally maintained by the TSF domain controllers. The key distribution effect is achieved because the user profile files can be downloaded from a TSF domain controller when the user logs onto a machine.

For RSA key generation, pseudo-random seeds are not put through the SHA-1 process before they are used in a RSA key generation as in the case of DSA or DH key generation. The details of the element types that contribute to a pseudo-random seed computation are documented in the NIST approved security policy documents associated with the above FIPS-140 certificates on the NIST web site. A pseudo-random value is used as the starting point of an array representing consecutive odd integers. The length of the array depends upon the size of the prime sought. The array is then sieved by finding multiples of each small prime (about a thousand small primes). Surviving candidates (starting from the smallest) are then each put through four iterations of Miller-Rabin until one is found that passes. If no prime is found, a new random starting value is selected and the process is repeated until a prime is found. Resetting the starting value when a new sieve is required limits the bias introduced by the sieving method.

The primality test for DH/DSA primes goes through one Miller Rabin and one Lucas. The mechanism to generate a seed for a DSA key pair is the same mechanism for which the crypto module has received the FIPS-140-2 DSA Certificate. The details of the element types that contribute to a seed computation are documented in the NIST approved security policy documents associated with the above FIPS-140 certificates on the NIST web site. Therefore, the entropy value for the mechanism is at least that of any FIPS-140 validated crypto modules. DSA key generation requires a pseudo-random seed to go through the SHA-1 process with all the mandatory checks for receiving the FIPS-140 certificates. The DSA prime values p and q are generated using the algorithm of FIPS-186-2 Appendix 2.2. The G value is generated using FIPS-186-2 Appendix 4.

SFR Mapping:

The **Cryptographic Protection** function satisfies the following SFRs:

- FCS_COP.1(a) – The TSF uses the DES or 3DES (56-bit or 168-bit 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.1(a) – (j) – See the Table 6-3 Cryptographic Standards and Evaluation Method
- FCS_CKM.1(a) – (e) - See the Table 6-3 Cryptographic Standards and Evaluation Method
- FCS_CKM.2, FCS_CKM.4 – See the Table 6-3 Cryptographic Standards and Evaluation Method
- FCS_CKM_EX.1 – The TSF validates keys by checking the hash.
- FCS_CKM_EX.2 – The TSF ensures the private keys are associated with the entities by encrypting the keys with a hash of the password of the key owner. The TSF stores private keys in PKCS#5 in a subdirectory with the user’s profile.

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 Access Control section) during which a web server request is serviced without prompting the client for identification and authentication, even though that client has been authenticated prior to making a web server request in the evaluated configuration.

6.1.4.1 Logon Type

The TOE supports six types of user logon: interactive (“Logon locally”), network (“Access this computer from Network”), batch (“Logon as a batch job”), service (“Logon as a service”), unlock (“Unlock screen saver”), and Network_ClearText (“Anonymous authentication to IIS”). The interactive logon type is for users who will be interactively using the system, such as a user being logged on at a workstation console. The network logon type is used when a user logs onto a remote network server to access resources. The batch logon type is intended for batch servers, where processes may be executing on behalf of a user without their direct intervention (e.g., COM - servers). The service logon type is used when a service process is started to provide a user context in which that service will operate. The unlock logon type is used when a user is forced to re-authenticate interactively after a specified time of inactivity. The network_clearText logon type is used when IIS is configured to not require a client requesting IIS services 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). For remote logon, a user must first logon on interactively for which a trusted path is provided (as described above). Additionally, the TSF uses IPsec 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).

Other actions that require the user to invoke the trusted path by using the **Ctrl+Alt+Del** key sequence and re-authenticate themselves are: initial user authentication with a smart card, changing passwords, and session unlocking (see section Session Locking Function).

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.

6.1.4.3 User Attribute Database

6.1.4.3.1 User and Group Accounts Definitions

Each TSF 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 associated 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; and,
- 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.

The actual composition of the user attribute database depends upon the type of TSF (e.g., stand-alone, domain member, DC). Specifically, the TOE allows the establishment of domains. Domains are used to allow a collection of TSFs to share a common set of policies and accounts. This is accomplished by establishing DCs that instantiate AD services (every TSF with the AD service is a DC) that define policies and accounts to be shared by TSFs in the domain. Note that *group policies* (see Security Management) can also be defined in the AD that apply to selected TSFs (e.g., systems) and accounts within the domain. If a TSF type is not a domain member, it will have only its own user attribute database. If a TSF type is a domain member, but not a DC, it will also have its own user attribute database. However, the policies and accounts of its DC will logically be included in that TSF’s user attribute database. If a TSF type is a DC, its user attribute database is defined within its AD and is generally shared with other TSFs in the domain.

In a domain, a user attribute database can be logically extended even further through trust relationships. Each DC can be configured to trust other domains. The result is that accounts from trusted domains can be used to access the trusting domain.

A forest is a set of one or more trees that do not form a contiguous namespace. The TSF allows a forest to enforce constraints on which users it trusts the other forest to authenticate. This 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. This cross-forest authentication enables secure access to resources when the user account is in one forest and the computer account is in another forest. A computer account is a user account where the user identity of the account is a computer identity belonging to a Windows domain.

6.1.4.3.2 Account Policies

Complimentary 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 account lockout policy with respect to interactive logons.

The password policy includes:

- The number of historical password to maintain 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; and,
- The minimum password length when changing to a new password (0-14).

The account lockout policy includes:

- Duration of the account lockout once it occurs;
- Number of failed logon attempts before the account will be locked out; and,
- The amount of time after which the failed logon count will be reset.

These policies allow the TSF to make appropriate decisions and change user attributes in the absence of an authorized administrator. For example, the TSF 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).

6.1.4.4 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 TSF (or NULL) is selected for the domain name, the local user account database is used. Otherwise the user account database on the target TSF’s DC 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, two types of logon may occur: NTLM or Kerberos. Kerberos is the default logon method and will be used if a Kerberos KDC is available. Generally, each DC includes a KDC in addition to its AD. If no KDC is available, NTLM will be used. 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 to the appropriate DC (or local TSF for a local account). The receiving TSF will compare the provided hashed password with the version 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 partially 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 yields an increment in 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 TSF 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 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.

Once a successful authentication occurs, the TSF will query its AD (via its DC), if applicable, for group policies relevant to the user that is attempting to logon. The TSF 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). The TSF 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).

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 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 in 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.

6.1.4.4.1 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.

6.1.4.4.2 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 four network security protocols, namely SSL versions 2.0 and 3.0, TLS version 1.0. 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.

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 RFC 2617 “HTTP Authentication: Basic and Digest Access Authentication.”

6.1.4.5 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 doesn't contain any client information. Identify allows the server to impersonate the client to perform access checks. Impersonate allows the server to impersonate the clients entire security context to access resources local to the server's TSF. Delegate allows the server to impersonate the client on local and remote TSFs.

6.1.4.6 Restricted Tokens

Whenever a process is created, or a thread is assigned an impersonation token, the TSF 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. The following pertains:

- 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.

6.1.4.7 Strength of Authentication

As indicated above, the TSF 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 as large as 14 characters. However, the administrator guide recommends that the minimum password length be configured to no less than eight (8)

characters (with at least 90 available characters, the password space is 4,304,672,100,000,000 available combinations). Therefore, in the evaluated configuration, the probability that a random attempt will succeed is less than one (1) in 5×10^{15} and the probability that, for multiple attempts within one minute, the probability that a random attempt will succeed is less than one (1) in 25×10^{12} .

During authentication, the TSF will not provide feedback that will reduce the probability before the metrics identified above. Furthermore, the TSF forces a delay between attempts, such that there can be no more than ten (10) attempts per minute.

For each subsequent failed logon following five (5) consecutive failed logon occurrences in the last 60 seconds, the Winlogon/Graphical Identification and Authentication (GINA) subcomponent sleeps for 30 seconds before showing a new logon dialog. It therefore supports the I&A function that no more than ten (10) interactive logon attempts are possible in any 60 second (one minute) 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.1 - The TSF locks the account after the administrator-defined threshold of unsuccessful logon attempts has occurred. The account will remain locked either until an authorized administrator unlocks it or until the duration defined by an authorized administrator has elapsed.
- FIA_ATD.1 - Each TSF has a user attribute database. 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.
- FIA_SOS.1 - The password and key spaces used by the TSF reduce the chance of guessing a password to less than one (1) in 2×10^{15} for a single random attempt and one (1) in 25×10^{12} for multiple attempts during a one minute period. The TSF can block a user from continuing to attempt to logon for a specified amount of time after a specified number of failed attempts such that there can be no more than ten attempts per minute and the TSF does not provide feedback during authentication that will reduce the probability of successfully guessing passwords.
- 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.7 - During an interactive logon, the TSF echoes the users password with "*" characters to prevent disclosure of the user's password.
- 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_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_EX - Each process and thread has an associated token that identifies the responsible user (used for audit and access), associated groups (used for access), privileges, 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.
- FTA_LSA_EX.1 - On a DC, accounts can be restricted to a workstation during a specific time and day. If the account has these restrictions, the members of the domain will then restrict the ability to logon to a system based upon the Logon Locally right (allows the user to interactively logon to given system), the time, and the day. If on a given system, the user can logon at a given time and

day, then the user will be allowed to logon and will be included in the groups assigned to that account and will have the privileges assigned to that account.

- FTA_MCS_EX.1 - Through locally logon right enforcement, accounts can be restricted to specific workstations 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.1(i) - An authorized administrator can define and modify a banner that will be displayed prior to allowing a user to logon.
- 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, session unlocking, and changing the user's password when the TSF requests/notifications (via the trusted path) the user to do so. 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. Additionally, IPSec is used to provide an additional trusted path for remote logons.
- FMT_SMR.3 - In order to assume the authorized administrator role (see the Security management Function), a user with one of the security-relevant administrative groups or security-relevant privileges must successfully logon. Furthermore, to switch between a user with privileged and an authorized administrator, a user must logoff and re-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 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 two logical roles: the *authorized administrator role* and the *authorized user role*.

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 of the several pre-defined administrative groups (e.g., Administrators and Backup Operators local group). 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 and is not in the administrator role (as defined above) is considered to be in an authorized user role.

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 passwords, 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 and 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.

6.1.5.3 Valid Password Attributes

The TSF ensures that only valid values are accepted as security attributes for the password. Valid values are values that 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 8 by the administrator. Subsequently, attempts to create passwords shorter than 8 will not be accepted by the TSF.

SFR Mapping;

The **Security Management function** satisfies the following SFRs:

- FMT_MOF.1(a): Only an authorized administrator can enable and disable the audit mechanism, select which audit event categories will be audited, and also select whether they will be audited for success and/or failure.
- FMT_MOF.1(b) - The TSF provides IPSec management functions that allow only an authorized administrator the ability to define if and how IPSec will be used to protect traffic amongst distributed TSFs.
- FMT_MSA_EX.2 - The TSF ensures that values for password security attributes meet the password complexity restrictions, if defined by the administrator.
- FMT_MTD.1(a) - Only an authorized administrator can clear the security event log. There are no interfaces to create or delete the security event log entries (see Audit Log Restricted Access Protection).
- FMT_MTD.1(b) - Only an authorized administrator can view the security event log. There are no interfaces to modify a security event (audit record) in the security event log (see Audit Log Restricted Access Protection).
- FMT_MTD.1(c) - 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 below) and private/public key pairs.
- FMT_MTD.1(d) - 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.1(e) - Only an authorized administrator can change the duration of lockouts.
- FMT_MTD.1(f) - Only an authorized administrator can change the minimum password length.
- FMT_MTD.1(h) - Only an authorized administrator can manage disk quotas and define actions to take when disk quotas are exceeded.
- FMT_MTD.1(l) - Only an authorized administrator can create, *change default, query, modify, delete, and clear* TSF data that is not considered audit records, user security attributes, authentication data and critical cryptographic security parameters (such as IPSec and EFS policy).
- FMT_MTD.1(m) - The TSF does not store passwords in clear text and does not provide any interfaces to read passwords.

- FMT_MTD.1(n) - The TSF allows only the authorized administrator to change the password complexity requirements.
- FMT_MTD.1(o) – The TSF allows a user to trigger the generation of a private/public key pair for their own account an authorized administrator may trigger the generation of a private/public key for any account.
- FMT_MTD.2 - Only an authorized administrator can specify and modify the maximum amount of failed logon attempts that may occur before the account is locked out.
- FMT_REV.1(a) - Only an authorized administrator can remove security attributes from users and group accounts. A procedure is described in the Administrator Guide that will instruct an authorized administrator on how to immediately remove security attributes from accounts.
- 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.
- FMT_SMF.1 - The TSF provides the administrator with the capability to modify the time and define the following policies: Audit Policy, Account Policy, Account Database Policy, User Rights Policy, Domain Policy, Group Policy, IPsec Policy, EFS Policy, Disk Quota Policy, DAC Policy, and the File Encryption Key Policy. Specifically, the TSF provides the administrator with the capability to perform the following:
 - DAC Policy
 - modify access control attributes associated with a named object
 - File Encryption Key Policy
 - delete encryption policy attributes associated with a file
 - 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
 - IPsec Policy
 - determine and modify the behavior of the function that protects data during transmission between parts of the TOE
 - 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
 - Account Database Policy
 - initialize and modify user security attributes
 - Disk Quota Policy
 - modify the quota settings on NTFS volumes
- 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 Protection provides:

- System Integrity;
- Internal TSF Transfer Protection;
- TSF Data Replication Consistency;
- Reference Mediation;
- Domain Separation;
- Abstract Machine Testing; and,
- Time Service.

6.1.6.1 System Integrity

The hardware platform included in the TOE is tested to ensure the security functions are supported. The tests are directed at determining correct operation of the central hardware components, such as the motherboard, as well as the set of attached peripheral devices, such as memory, disks, video, I/O ports, etc. Specifically, these test are designed to ensure that the features most directly relied upon to support the security functions are operating correctly (i.e., interrupt handling, memory management, task management, privileged instructions).

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 (ISAKMP/Oakley) 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 ESP to provide data confidentiality for IP packets. ESP performs encryption using the DES CBC algorithm and 3DES CBC. In addition to ESP, the TSF implements IP AH. AH provides integrity, authentication and anti-replay. AH uses a hashing algorithm, such as SHA-1, to compute a keyed message hash for each IP packet.

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 together 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 and authentication algorithms that are agreed upon by the two computers during security negotiations.

Phase I (Main Mode SA) Key Negotiation: During the first phase, the two computers establish a secure, authenticated channel. This is called the Phase I or main mode SA. IKE automatically provides necessary identity protection during this exchange.

- Policy Negotiation: To negotiate a policy the following four mandatory parameters are negotiated as part of the main mode SA:
 - The encryption algorithm (DES or 3DES) ;
 - The integrity algorithm (MD5 or SHA1);
 - The DH group to be used for the base keying material: Group 1 (768 bits of keying material) Group 2 (1,024 bits), or Group 2048 (2,048 bits)
 - The authentication method (Kerberos V5, certificate, or preshared key authentication)
 - If certificates or preshared keys are used for authentication, the computer identity is protected. If Kerberos V5 authentication is used, the computer identity is unencrypted until encryption of the entire identity payload takes place during authentication.
- DH exchange (of public values) - Only the base information required by the DH key determination algorithm to generate the shared, secret key is exchanged. After this exchange, the IKE service on each computer generates the master key that is used to protect authentication.
- Authentication: To prevent a successful man-in-the-middle attack, the computers attempt to authenticate the DH exchange. Without successful authentication, communication will not proceed.

Phase II (Quick Mode SA) Key Negotiation: In this phase, SAs are negotiated on behalf of the IPSec driver. The following are the steps that comprise a quick mode negotiation.

- Policy negotiation occurs. The IPSec computers exchange the following requirements for securing the data transfer:
 - The IPSec protocol (AH or ESP)
 - The hash algorithm for integrity and authentication (MD5 or SHA1)
 - The algorithm for encryption, if requested (3DES or DES)
- A common agreement is reached, and two SAs are established. One SA is for inbound communication and the other is for outbound communication.

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 snapin 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.

For outbound IP traffic, the processing occurs as follows:

- The IP stack calls into the IPsec module to apply security action (e.g. encrypt) to the packet based on filters defined in policy.
- If this is the first packet processed for the specified source/destination pair, a set of security parameters are retrieved and/or generated via the Internet standard based key management protocol (ISAKMP). The parameters include the security context (used to establish common keys with the destination machine and enforce particular policy variations), generated keys, and others, such as the specific algorithm, which are mapped to a structure known as a SA.
- IPsec action is performed (encrypt, AH, etc.).

For inbound IP traffic, the processing occurs as follows:

- The IP stack calls the IPsec module to perform security action (decrypt, authenticate).
- The IPsec module processes the next header to determine what, if any, security service has been applied (e.g., ESP, AH).
- If a security service has been applied, then the IPsec module retrieves the appropriate parameters via the SA to process the packet.
- The packet is processed by obtaining the appropriate algorithm to process the security action (decrypt, verify signature (ESP integrity and AH integrity)) and removes security specific headers. Note that ESP integrity and encryption can only be chosen together.

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).

6.1.6.3 TSF Data Replication Consistency

In general, directory data reside in more than one place on the network. They also need to be equally useful to all users. 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. 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 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.

The AD service allows for specific data to be replicated within the TOE. The AD namespace includes a domain tree structure and a forest 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 returns the location of an object based on an object attribute provided by the user.

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.

A tree is a set of one or more Windows 2003/XP 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.

A GC Server is a DC that stores specific information about all objects in the 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.

Enterprises can be a single-tree or a multi-tree. Naming within a given tree is always contiguous.

There are two types of TSF data replicated consistently throughout the TOE. They consist of Group Policy Objects (GPOs) and 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 you add, remove, or modify 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 fourth 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, replicated to every domain controller in that domain, but not beyond its domain.
- **Schema partition:** all object types (w/attributes) that can be created in AD, 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, 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 partial replica of the domain directory partition for all other domains in the enterprise, contains a subset of the properties for all objects in all domains in the enterprise. (Is read-only)

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). Currently in Windows 2003/XP there are five 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.
- Primary DC (PDC) emulator - a Windows 2003/XP DC that advertises itself as the PDC to member servers and domain controllers.
- 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 three 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, by time stamp if the version numbers are the same, or 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 can be discarded.

Note that all reconciliation operations are logged, and authorized administrators have the option of recovering and using the rejected values.

6.1.6.4 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 ensure 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 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 what access rights to the object the subject was granted to the object according to the security policy. On every attempt to use a handle, the TSF ensure 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).

6.1.6.5 Domain Separation

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; and,
- 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, etc.), 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.

The evaluated configuration is restricted to devices that do not allow for the loading of executable data.

6.1.6.6 Abstract Machine Testing

During the evaluation of the TOE, tests were executed to demonstrate the hardware mechanisms included in the TOE perform correctly to support the SFs.

6.1.6.7 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).

SFR Mapping:

The **TSF Protection function** satisfies the following SFRs:

- **TRANSFER_PROT_EX.1** – 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.
- **FPT_TRC_EX.1** – The TSF provides consistency of replicated GPOs and DS data by implementing a well-defined TSF replication algorithm.
- **TRANSFER_PROT_EX.3** - The TSF implements IP AH. AH provides integrity, authentication and anti-replay. AH uses a hashing algorithm, such as SHA-1, 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).
- **FPT_RPL_EX.1** – The TSF implements IP AH. AH provides integrity, authentication and anti-replay. AH uses a hashing algorithm, such as SHA-1, to compute a keyed message hash for each IP packet. The TSF may reject the packet or audit the event if the IPSec service results challenge the integrity of the packet.
- **FPT_RVM.1** – The TSF provides reference mediation of all the objects covered by the DAC policy. Reference mediation is primarily enforced through handle enforcement. Once an access policy decision is made by the TSF, this policy is enforced via the handle enforcement checks applied every time a handle is used. In this manner, access to objects is assured to be consistent with the security policy even though the security policy is not checked on all use of an object. Some objects are directly accessed by name without obtaining a handle first. In these cases, the TSF checks the request against the access policy directly.
- **FPT_SEP.1** – The TSF provides a security domain to protect itself through hardware, the processor kernel mode, controlled state-transitions, process isolation, and memory protection. Processes are managed by the TSF kernel-mode software and have private address spaces and process context.
- **FPT_STM.1, FMT_MTD.1(g)** - The real-time clock in each Windows 2003/XP 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_AMT.1** - Tests were available during the evaluation that demonstrated the correct operation of the hardware mechanisms included in the TOE.

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. 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, 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 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 2003/XP 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 2003/XP 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_MOF.1(c) - Only the authorized user and an authorized administrator can unlock a locked session.
- FMT_MTD.1(k) - 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.

6.2 TOE Security Assurance Measures

The following assurance measures are applied to Windows 2003/XP to satisfy the CC EAL4 assurance requirements:

- Process Assurance;
- Delivery and Guidance;
- Design Documentation;
- Tests; and,
- Vulnerability Assessment.

6.2.1 Process Assurance

6.2.1.1 Configuration Management

The Configuration Management (CM) measures applied by Microsoft ensure that Configuration Items (CIs) are uniquely identified, and that documented procedures are used to control and track changes that are made to the TOE. Microsoft ensures changes to the implementation representation are controlled with the support of automated tools and that TOE associated CI modifications are properly controlled. Microsoft performs CM on the TOE implementation representation, design, tests, user and administrator guidance, the CM documentation, lifecycle documentation, vulnerability analysis, and security flaws. Microsoft documents and follows an acceptance plan for how CIs are approved. Microsoft ensures that the TOE is uniquely referenced and labeled with its reference. Microsoft uses, and documents how they use, automated tools to support TOE generation. These activities are documented in the Windows 2003/XP CM Manual.

Microsoft applies procedures to accept and act upon reported security flaws and requests to correct security flaws. Microsoft designates specific points of contact for user reports and security related inquiries. The procedures are documented and describe how security flaws are tracked, that for each security flaw a description and status of the correction of the security flaw is provided, that corrective actions are identified for each security flaw, how flaw information is provided (corrective actions and guidance on corrective actions). The procedures ensure that all reported flaws are corrected and that corrections are issues to TOE users, and that the flaws do not introduce new flaws. The procedures also ensure a timely response to reported flaws and the automatic distribution of security flaw reports to the affected users. These activities are documented in the Windows 2003/XP CM Manual.

6.2.1.2 Life-Cycle Support

Microsoft ensures the adequacy of the procedures used during the development and maintenance of the TOE through the use of a comprehensive life-cycle management plan. Microsoft includes security controls on the development environment that are adequate to provide the confidentiality and integrity of the TOE design and implementation that is necessary to ensure that secure operation of the TOE. Microsoft achieves this through the use of a documented model of the TOE life cycle and well-defined development tools that yield consistent and predictable results. Additionally, Microsoft documents the implementation dependent options and the meaning of all statements used in the implementation. This information and these procedures are documented in the Windows 2003/XP Life Cycle Management Plan.

SAM Mapping:

The **Process assurance measure** satisfies the following SARs:

- ACM_AUT.1;
- ACM_CAP.4;
- ACM_SCP.2;
- ALC_DVS.1;
- ALC_FLR.3;
- ALC_LCD.1; and,
- ALC_TAT.1.

6.2.2 Delivery and Guidance

Microsoft provides delivery documentation and procedures to identify the TOE, allow detection of unauthorized modifications of the TOE and installation and generation instructions at start-up. Microsoft's delivery procedures describe the electronic and non-electronic procedures to be used to detect modification to the TOE. The installation and generation procedures describe the steps necessary to place Windows 2003/XP into the evaluated configuration. These procedures are documented in the Windows 2003/XP Delivery and Operation Procedures.

Microsoft provides administrator and user guidance on how to perform the TOE security functions and warnings to authorized administrators and users about actions that can compromise the security of the TOE. Administrator and User guidance is documented in the Windows 2003/XP Administrator Guide

SAM Mapping:

The **Delivery and Guidance assurance measure** satisfies the following SARs:

- ADO_DEL.2;
- ADO_IGS.1;
- AGD_ADM.1; and,
- AGD_USR.1.

6.2.3 Design Documentation

The Windows 2003/XP "Security Design Documentation" is an extensive set of documents describing all aspects of the TOE security design, architecture, mechanisms, and interfaces. The Security Design Documentation consists of a large number of related documents. These documents are:

- Introduction: Describes the form, content, and organization of the System Design documentation.
- Security Policy: Provides an informal description and model of the access control policy for the system.
- System Decomposition Summary: This document describes the decomposition of the system and identifies the subsystems in terms of components.

- Component Descriptions (several): There are several of these documents; one each for the system components defined in the Decomposition Summary document. Each document describes the component and identifies the modules within the component in terms of subcomponents.
- Subcomponent Designs (many): There are many of these documents; one each for the subcomponents defined in the several Component Description documents. Each subcomponent design document presents the following:
 - Summary identifying the subcomponent's name, implementation location, and execution environment.
 - A description of the design of the subcomponent and a summary of its security functions and mechanisms.
 - A specification of each TSF interface implemented by the subcomponent. The following is provided for each TSF interface: purpose, parameters, security checks, and security effects.
 - A correspondence matrix that identifies for each TSF interface, which security functions the interface's checks and effects help implement. The matrix includes a rationale for this correspondence.
 - A test family summary that describes test cases implemented in the security tests for each API.

SAM Mapping:

The **Design Documentation assurance measure** satisfies the following SARs:

- ADV_FSP.2 - The sum of all TSF interface specifications from each of the Subcomponent Design documents fully describes all interfaces to the TSF.
- ADV_HLD.2 - The system components satisfy the requirement for decomposing the TOE into subsystems. Each component corresponds to a subsystem. The Component Decomposition Summary document and all of the Component Description documents fully describe each component.
- ADV_IMP.1 - The source code used to generate the TOE satisfies this requirement.
- ADV_LLD.1 - The subcomponents, which are a further decomposition of the components, satisfy the requirement to decompose each subsystem into module. Each subcomponent is a module. The design descriptions and TSF interface specifications from each of the Subcomponent Design documents fully describes each subcomponent.
- ADV_SPM.1 - The Security Policy document fully presents an informal security model for the TOE.
- ADV_RCR.1 - Most of the correspondence between the various design documentation is implicit to the way in which the documentation is structured. The way that this correspondence is evident within the design documentation is:
 - ST-TSS to FSP - This is the principal explicit correspondence provided within the Security Design documentation. This correspondence is captured in all the TSF interface correspondence matrices from each of the Subcomponent Design documents.
 - FSP to HLD - Since the FSP is presented on a per-subcomponent basis, this correspondence is implicit since each Component Description document explicitly identifies which subcomponents (and hence which TSF interfaces) are contained within each Component.
 - HLD to LLD - As above, the Component Description documents explicitly identify the association between components and subcomponents.

- LLD to IMP - The summary information for each Subcomponent Design document identifies the location within the TOE source code tree where that subcomponent implementation is contained.

6.2.4 Tests

The TOE test documentation has been created to demonstrate appropriate breadth and depth of coverage. The test documentation describes how all security relevant APIs are tested, specifically describing all test cases and variations necessary to demonstrate that all security checks and effects related to the API are correctly implemented. The test documentation provides correspondence between the security-relevant APIs and applicable tests and test variations. The test documentation describes the actual tests, procedures to successfully execute the tests, and expected results of the tests. The test documentation also includes results in the form of logs resulting from completely exercising all of the security test procedures.

The test documentation consists of four parts: a *test plan* (“Windows 2003/XP Security Test Plan”), *test families*, *test suites*, and *test results*.

- The test plan describes the form, content, and organization of test documentation. It also summarizes each of the test suites and includes high-level procedures for exercising the tests.
- The test families described the set of security-relevant test cases on a per-subcomponent basis. These descriptions include references to the corresponding test suites that implement those test cases. Note that every test case corresponds to at least one test suite.
- The test suites include both documentation and an actual implemented test (if applicable). Test suites are organized around tests that share a common theme, such as handle enforcement, privilege enforcement, auditing, etc. The test suite documentation describes the purpose and “theme” for the test suite, the set of test variations that are exercised for each of its corresponding test cases, procedures to successfully exercise the test suite, and the expected results. The test suite documentation also implicitly includes the actual tests that provide specific details regarding test variations and expected results.
- The test results are essentially the set of logs resulting from completely exercising all of the security test procedures. These logs include summaries of the results in terms of total test variations, counts of variations that passed, failed, or blocked (i.e., were unable to run), and detailed information about each variation that was attempted, including more detailed results and expected results.

SAM Mapping:

The **Tests assurance measure** satisfies the following SARs:

- ATE_COV.2 - The set of test families describe the test cases for each of the security-relevant interfaces of the TOE. The test families indicate which test suites (and therefore which tests) are used to satisfy the test cases identified for each interface.
- ATE_DPT.1 - The test suites include test variation descriptions that demonstrate that all of the corresponding test cases (and therefore security checks and effects) are appropriately exercised.
- ATE_FUN.1 - Together, the test documents describe the security functions to be tested, how to successfully test all of them, the expected results, and the actual test results after exercising all of the tests.
- ATE_IND.2 - The TOE and test suites will be available for independent testing.

6.2.5 Vulnerability Assessment

6.2.5.1 Evaluation of Misuse

The administrator guidance documentation describes the operation of Windows 2003/XP and how to maintain a secure state. The administrator guide also describes all operating assumptions and security requirements outside the scope of control of the TOE. The administrator guidance documentation has been developed to serve as a complete, clear, consistent, and reasonable administrator reference. This administrator guidance documentation is documented in:

- The Windows 2003/XP Administrator Guide

The misuse analysis shows that the administrative guidance completely addresses managing the TOE in a secure configuration.

- The Windows 2003/XP Vulnerability Analysis

6.2.5.2 Strength of TSFs and Vulnerability Analysis

The strength of TSF analysis demonstrates that the SOF claims made in the ST for all probabilistic or permutation mechanisms are correct. Microsoft performs a systematic vulnerability analyses of the TOE to identify weaknesses that can be exploited in the TOE. Microsoft documents the status of identified vulnerabilities and demonstrates that for each vulnerability, the vulnerability cannot be exploited in the intended environment and that the TOE is highly resistant to obvious penetration attacks. The SOF and vulnerability analysis are documented in:

- The Windows 2003/XP Vulnerability Analysis

SAM Mapping:

The **Vulnerability Assessment assurance** measure satisfies the following SARs:

- AVA_MSU.2;
- AVA_SOF.1; and,
- AVA_VLA.2.

7. Protection Profile Claims

This section provides the PP conformance claim statements and supporting justifications of conformance with the CAPP.

7.1 CAPP Conformance Claim Reference

The TOE conforms to the Controlled Access Protection Profile (CAPP), Version 1.d, National Security Agency, 8 October 1999.

7.1.1 CAPP Requirements in ST

The CAPP requirements included in this ST are identified in Section 5. For each CAPP requirement included in this ST, Section 5 also indicates what operation, if any has been performed. The specific operations that were performed are highlighted in Section 5 as part of the requirement statements.

7.1.2 CAPP Differences and Enhancements

The following list in Table 7-1 clearly identifies the delta between this ST and the CAPP with respect to threats, assumptions, policies, objectives, SFRs and assurance requirements. The ST has primarily added additional items, or in the case of assurance requirements, enhanced requirements from EAL 3, as required in the CAPP, to EAL 4. This section categorizes the delta into differences and enhancements. Differences are considered changes to the PP content. Enhancements are considered the addition of new items or the replacement of an item in the CAPP with a higher hierarchical item. This section provides rationale that each difference and enhancement complies with CAPP and does not introduce any inconsistencies.

Table 7-1 also indicates when a requirement that is included in the CAPP has changed due to an International Interpretation, and is therefore different as presented in this ST. These requirements are identified in Table 7-1 by the word "Interpreted" in the Modification column. Note that these requirements are denoted in Section 5 by an italicized parenthetical following those changed requirement elements (e.g. *(per International Interpretation #51)*)

Table 7-1 CAPP Modifications

Category	Name	Modification
Threat	T.AUDIT_CORRUPT	Addition
Threat	T.CONFIG_CORRUPT	Addition
Threat	T.OBJECTS_NOT_CLEAN	Addition
Threat	T.SPOOF	Addition
Threat	T.SYSACC	Addition
Threat	T.UNAUTH_ACCESS	Addition
Threat	T.UNAUTH_MODIFICATION	Addition
Threat	T.UNDETECTED_ACTIONS	Addition
Threat	T.USER_CORRUPT	Addition
Threat	T.ADMIN_ERROR	Addition
Threat	T.AUDIT_COMPROMISE	Addition
Threat	T.EAVESDROP	Addition

Category	Name	Modification
Threat	T.MASQUERADE	Addition
Threat	T.POOR_DESIGN	Addition
Threat	T.POOR_IMPLEMENTATION	Addition
Threat	T.REPLAY	Addition
Threat	T.UNATTENDED_SESSION	Addition
Threat	T.UNIDENTIFIED_ACTIONS	Addition
Threat	T.ADDRESS_MASQUERADE	Addition
Threat	T.TCPIP_ATTACK	Addition
Policy	P.AUTHORIZATION	Addition
Policy	P.ADD_IPSEC	Addition
Policy	P.WARN	Addition
Objective	O.AUDIT_PROTECTION	Addition
Objective	O.PROTECT	Addition
Objective	O.TRUSTED_PATH	Addition
Objective	O.LEGAL_WARNING	Addition
Objective	O.LIMIT_AUTHORIZATION	Addition
Objective	O.ENCRYPTED_DATA	Addition
Objective	O.IPSEC	Addition
Objective	O.ASSURANCE	Addition
Objective	O.MEDIATE	Addition
SFR	FAU_GEN.1	Refinement
SFR	FAU_SAR.1	Refinement
SFR	FAU_STG.1	Interpreted, Refinement
SFR	FAU_STG.4	Refinement
SFR	FCS_COP.1(a) thru (j)	Addition
SFR	FCS_CKM.1(a) thru (e)	Addition
SFR	FCS_CKM.2	Addition
SFR	FCS_CKM.4	Addition
SFR	FCS_CKM_EX.1	Addition
SFR	FCS_CKM_EX.2	Addition
SFR	FDP_ACC.2(a)	Addition
SFR	FDP_ACC.2(b)	Addition
SFR	FDP_ACC.2(c)	Addition
SFR	FDP_ACF.1(a)	Refinement

Category	Name	Modification
SFR	FDP_ACF.1(b)	Addition
SFR	FDP_ACF.1(c)	Addition
SFR	FDP_IFC.1(a)	Addition
SFR	FDP_IFC.1(b)	Addition
SFR	FDP_IFF.1(a)	Addition
SFR	FDP_IFF.1(b)	Addition
SFR	FDP_ITT.1	Addition
SFR	FDP_UCT.1	Addition
SFR	FDP_UIT.1	Addition
SFR	FIA_AFL.1	Addition
SFR	FIA_SOS.1	Refinement
SFR	FIA_UAU.6	Addition
SFR	FIA_USB.1_EX.1	Refinement
SFR	FMT_MOF.1(a)	Addition
SFR	FMT_MOF.1(b)	Addition
SFR	FMT_MOF.1(c)	Addition
SFR	FMT_MOF.1(d)	Addition
SFR	FMT_MSA.1(b)	Addition
SFR	FMT_MSA.1(c)	Addition
SFR	FMT_MSA.1(d)	Addition
SFR	FMT_MSA.1(e)	Addition
SFR	FMT_MSA.1(f)	Addition
SFR	FMT_MSA_EX.2	Addition
SFR	FMT_MSA.3(b)	Addition
SFR	FMT_MSA.3(c)	Addition
SFR	FMT_MSA.3(d)	Addition
SFR	FMT_MSA.3(e)	Addition
SFR	FMT_MTD.1(e) thru (o)	Addition
SFR	FMT_MTD.2	Addition
SFR	FMT_SAE.1	Addition
SFR	FMT_SMF.1	Addition
SFR	FMT_SMR.3	Addition
SFR	TRANSFER_PROT_EX	Addition
SFR	FPT_TRC_EX.1	Addition

Category	Name	Modification
SFR	TRANSFER_PROT_EX.3	Addition
SFR	FPT_RPL_EX.1	Addition
SFR	FRU_RSA.1	Addition
SFR	FTA_LSA_EX.1	Addition
SFR	FTA_MSC_EX.1	Addition
SFR	FTA_SSL.1	Addition
SFR	FTA_SSL.2	Addition
SFR	FTA_SSL.3	Addition
SFR	FTA_TAB.1	Addition
SFR	FTA_TSE.1	Addition
SFR	FTA_TRP.1	Addition
SAR	ACM_AUT.1	Addition for EAL4
SAR	ACM_CAP.4	Upgrade for EAL4
SAR	ACM_SCP.2	Upgrade for EAL4
SAR	ADO_IGS.1	Interpreted
SAR	ADO_DEL2	Upgrade for EAL4
SAR	ADV_FSP.2	Upgrade for EAL4
SAR	ADV_IMP.1	Upgrade for EAL4
SAR	ADV_LLD.1	Addition for EAL4
SAR	ADV_SPM.1	Addition for EAL4
SAR	ALC_FLR.3	Augment to EAL4
SAR	ALC_LCD.1	Addition for EAL4
SAR	ALC_TAT.1	Addition for EAL4
SAR	ATE_COV.2	Addition for EAL4
SAR	AVA_MSU.2	Upgrade for EAL4

7.1.2.1 Threat Enhancements

The CAPP does not identify specific threats that are to be addressed by a compliant TOE. The ST includes specific threats to help readers understand the types of attacks that the TOE can address. These threats apply to aspects of the TOE that are included in the CAPP as well as additional TOE features presented in this ST.

7.1.2.2 Policy Enhancements

The ST includes three additional organizational policies from the CAPP, which the TOE addresses. One of these policies reflects an optional policy, which the TOE can support, depending upon configuration settings identified in Guidance Documents. The optional policy reflects the TOE ability to provide IPsec. Since IPsec may not be appropriate for all deployments of the TOE, it is included in the ST as an optional policy, P.ADD_IPSEC. This TSF implementation of IPsec is discussed in the TSS, corresponds to a

functional security requirement, which in turn supports the P.ADD_IPSEC organizational policy. Including the IPsec policy in the ST, complements the CAPP policies.

The remaining policies reflect other areas where the TOE includes functionality that is beyond that specified in the CAPP. The additional functionality and corresponding supported policies are fully compatible with the CAPP.

7.1.2.3 Objective Enhancements

The additional objectives in the ST reflect additional functionality and detail that was not included in the CAPP. These objectives generally are a result of the additional material (e.g., threats and policies) used to characterize the environment. The Rationale, Section 8 provides traceability between objectives and requirements.

7.1.2.4 SFR Enhancements

The additional SFRs reflect additional functionality that the TOE provides to meet the security objectives for the environment that is characterized in the ST. The additional SFRs are compatible with the CAPP. FDP_ACC.2(a) is included in this ST and is hierarchical to the CAPP requirements FDP_ACC.1. As indicated in Table 7-1 six requirements in the CAPP were further refined. These requirements are FAU_GEN.1, FAU_SAR.1, FAU_STG.1, FAU_STG.4, FDP_ACF.1(a), and FIA_SOS.1. These refinements are described below and these requirements remain compliant with the CAPP.

- FAU_GEN.1 is refined further than the CAPP to specify the audit events that are related to SFRs that are not included in the CAPP. The CAPP FAU_GEN.1 requirement includes the statement that the events listed meet the basic level of audit, with the exception of FIA_UID.1's user identity during failures. The events listed in the FAU_GEN.1 requirement in this ST is a superset of the events listed in the CAPP FAU_GEN.1 requirement. The additional events are related to the additional SFRs included in this ST that are not in the CAPP, however, these additional events are not at the basic level of audit. The refinements made in the FAU_GEN.1 requirement in this ST are to clarify the distinction between the audit events that are included for CAPP compliance and those that are added beyond the CAPP and to clarify that the additional audit events are not claimed to be at any specified level of audit.
- FAU_SAR.1 is refined to restrict the ability to view the audit records to only the authorized administrator and to provide the authorized administrator with a tool to access the audit records.
- FAU_STG.1 is refined to make a stronger claim of protection of the audit records by requiring that the audit records be protected from all modification, by removing the ability to perform "authorized" modifications.
- FAU_STG.4 is refined only to allow for a more readable requirement.
- FDP_ACF.1(a) is refined further than the CAPP to add additional security attributes associated with a subject that the DAC policy is based upon.
- FIA_SOS.1 is refined further than the CAPP to require a stronger secret than that specified in the CAPP and also to require a delay between authentication attempts.
- FIA_USB.1_EX (which is labeled FIA_USB.1 in the CAPP but is an explicit requirement) is refined to ensure the user identity associated with auditable events is unique and to allow for the association of a maximum resource quota to subjects acting on behalf of users.

7.1.2.5 Security Assurance Requirement Enhancements

The ST has upgraded and added additional security assurance requirements to reflect that the assurance measures in place for the TOE are at EAL 4 and augmented with ALC_FLR.3 (Systematic Flaw Remediation). The ST augmented EAL 4 is an appropriate claim as discussed in the rationale section 8.x.

The CAPP requires EAL 3. Since EAL 4 augmented is hierarchical to EAL 3, the SAR upgrades still fully comply with the assurance requirements in the CAPP.

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;
- Security Assurance Requirements;
- TOE Summary Specification;
- Security Functional Requirement Dependencies; and,
- Internal Consistency.

8.1 Security Objectives Rationale

This section shows that all threats, secure usage assumptions, and organizational security policies are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption, organizational security policy, or threat. Tables 8.1 and 8.2 present the mapping of objectives to the security environment.

8.1.1 TOE IT Security Objectives Rationale

This section provides evidence demonstrating the coverage of threats and organizational policies by the IT security objectives. The following table shows the threats and organizational policies that each IT security objective addresses.

Table 8-1 IT Security Objectives Rationale Mapping

IT Security Objectives	Threats and Organizational Policies
O.AUTHORIZATION	T.SYSACC T.MASQUERADE T.UNATTENDED_SESSION P.AUTHORIZED_USERS T.UNAUTH_ACCESS
O.DISCRETIONARY_ACCESS	T.USER_CORRUPT P.NEED_TO_KNOW
O.AUDITING	T.UNDETECTED_ACTIONS T.AUDIT_COMPROMISE P.ACCOUNTABILITY T.UNIDENTIFIED_ACTIONS
O.AUDIT_PROTECTION	T.AUDIT_CORRUPT T.AUDIT_COMPROMISE
O.RESIDUAL_INFORMATION	P.NEED_TO_KNOW T.OBJECTS_NOT_CLEAN

IT Security Objectives	Threats and Organizational Policies
O.MANAGE	P.ACCOUNTABILITY P.AUTHORIZED_USERS P.NEED_TO_KNOW T.UNIDENTIFIED_ACTIONS
O.ENFORCEMENT	P.ACCOUNTABILITY P.AUTHORIZED_USERS P.NEED_TO_KNOW P.ADD_IPSEC
O.PROTECT	T.UNAUTH_MODIFICATION T.CONFIG_CORRUPT T.USER_CORRUPT T.AUDIT_COMPROMISE T.UNAUTH_ACCESS
O.TRUSTED_PATH	T.SPOOF T.REPLAY
O.LEGAL_WARNING	P.WARN
O.LIMIT_AUTHORIZATION	P.AUTHORIZATION
O.IPSEC	P.ADD_IPSEC T.EAVESDROP T.REPLAY
O.ENCRYPTED_DATA	T.USER_CORRUPT T.UNAUTH_ACCESS
O.ASSURANCE	T.ADMIN_ERROR T.POOR_DESIGN T.POOR_IMPLEMENTATION T.UNATTENDED_SESSION T.UNIDENTIFIED_ACTIONS T.TCPIP_ATTACK
O.MEDIATE	T.ADDRESS_MASQUERADE

The following objectives are sufficient to address all of the threats and organizational policies in the ST.

O.AUTHORIZATION – Ensuring that the TOE and its resources are protected from unauthorized access counters the threats T.UNAUTH_ACCESS and T.SYSACC since the execution of these threats relies upon unauthorized access to the TOE. T.MASQUERADE is also mitigated by this objective because it ensures that only authorized users are allowed access to a resource. Additionally, this objective implements the policy P.AUTHORIZED_USER by ensuring that only authorized users gain access to the TOE and its

resources. T.UNATTENDED_SESSION is mitigated by ensuring the TOE does not allow unauthorized access to the TOE and its resources.

O.DISCRETIONARY_ACCESS – By ensuring that authorized users can define which users can access their resources, the threat T.USER_CORRUPT is countered because the TSF enforces the authorized users' restrictions thus preventing users from accessing data not allowed by the user authorized to restrict access to that data. This objective ensures that the TSF enforces the restrictions to resources defined by the authorized users, thereby implementing the policy P.NEED_TO_KNOW.

O.AUDITING – By ensuring that the TSF record security relevant actions of users and present them to the authorized administrator, the threat T.UNDETECTED_ACTIONS is countered because the record of actions produced by the TSF will ensure that unauthorized actions will not go undetected.

T.AUDIT_COMPROMISE is mitigated by this objective because the objective ensures that the generation of audit data can not be prevented by unauthorized users. This objective ensures that a record of actions is produced and made available to the authorized administrator thereby implementing the policy P.ACCOUNTABILITY by providing the ability to review actions of individuals on the TOE and to hold them accountable for their actions. T.UNIDENTIFIED_ACTIONS is mitigated by ensuring the TOE present audit data to the authorized administrator.

O.AUDIT_PROTECTION – By ensuring that the audit information is protected, the threats T.AUDIT_CORRUPT and T.AUDIT_COMPROMISE are countered because unauthorized access will be prevented and audit information will not be lost or tampered with by unauthorized users.

O.RESIDUAL_INFORMATION – By ensuring that information in a protected resource is not released when the resource is recycled, the threat T.OBJECTS_NOT_CLEAN is countered because the TSF will always remove data from resources between uses by different users. This objective supports the policy P.NEED_TO_KNOW because it enforces the restrictions on resources defined by authorized users by ensuring that information is not left behind in a resource that may have different restrictions placed upon it.

O.MANAGE – By ensuring that all the functions and facilities necessary to support the authorized administrator in managing TOE security are provided, support is provided to implement the P.ACCOUNTABILITY, P.AUTHORIZED_USERS, and P.NEED_TO_KNOW policies because it requires the system to provide functionality to support the management of audit, resource protection, and system access protection. T.UNIDENTIFIED_ACTIONS is mitigated by ensuring the TOE offers the necessary management functions for the authorized administrator to securely manage the TOE.

O.ENFORCEMENT – By ensuring that organizational policies are enforced, the policies P.ACCOUNTABILITY, P.AUTHORIZED_USERS, P.ADD_IPSEC, and P.NEED_TO_KNOW are supported because the objective ensures that functions are invoked and operate correctly.

O.PROTECT – By ensuring that the TSF protects itself including its data and resources from external tampering, the threats T.UNAUTH_ACCESS and T.CONFIG_CORRUPT are countered. Additionally, support to counter the threats T.USER_CORRUPT, T.UNAUTH_MODIFICATION and T.AUDIT_COMPROMISE are supported. Ensuring that unauthorized access to the TSF data and resources is prevented disallows the above threats from being executed since they rely upon unauthorized access to TSF data or the modification of the TSF to a state where the security functions are not enforced thereby ensuring that the TSF is never bypassed.

O.TRUSTED_PATH – By ensuring that there is a capability to allow users to ensure they are communicating with the TSF during initial user authentication, the threat T.SPOOF is countered because the execution of the threat relies upon the ability to masquerade as the TSF. Countering T.REPLAY is supported in that authentication data cannot be captured by an authorized entity.

O.LEGAL_WARNING – By ensuring that users are aware of legal issues involving use of the TOE before access to resource is allowed implements the policy P.WARN because it provides the users with a warning of the ramifications of unauthorized use of the TOE.

O.LIMIT_AUTHORIZATION – By providing a capability to limit the extent a user's authorizations, the policy P.AUTHORIZATION is implemented because each user's authorizations can be limited.

O.IPSEC – By ensuring that the a capability is provided to protect system data in transmission between separate parts of the TOE, the policy P.ADD_IPSEC is implemented because it requires the system to provide this capability to protect system data in transmission between distributed parts of the TOE. By protecting data during transmission data cannot be intercepted allowing the TOE to mitigate T.EAVESDROP. The mitigation of T.REPLAY is assisted by ensuring data during transmission is protected from capture and resubmission.

O.ENCRYPTED_DATA – By ensuring that only users that encrypted data may receive that data decrypted the threat T.USER_CURRUPT and T.UNAUTH_ACCESS are countered because access to decrypted data from a user other than the user that encrypted the data is prevented

O.ASSURANCE – By ensuring that the guidance documentation is accurate and not misleading the threat that the TOE is incorrectly installed or configured, T.ADMIN_ERROR, is countered. The application of sound design principles and techniques, functional testing, and penetration testing mitigate the threats T.POOR_DESIGN and T.POOR_IMPLEMENTATION that errors exist in the TOE design and implementation. T.UNATTENDED_SESSION and T.UNIDENTIFIED_ACTIONS are mitigated by ensuring there is sufficient guidance to users and authorized administrators with respect to using the security functions. T.TCPIP_ATTACK is mitigated by ensuring the TOE has undergone a vulnerability analysis and penetration testing which will ensure the TOE is sufficiently robust to protect itself against published exploits.

O.MEDIATE – By ensuring that all network packets that flow through the TOE are subject to the information flow policies, a user cannot modify the identification TOE interface associated with them which mitigates the threat T.ADDRESS_MASQUERADE.

All of the organizational policies and threats are addressed by the IT security objectives. For each policy and threat, the associated IT security objectives are appropriate to address each policy and threat associated with them in Table 8.1. Given that the IT Security Objectives are met, the organizational policies will be implemented and the threats will be countered.

8.1.2 Non-IT Security Objectives for the Environment Rationale

This section provides evidence demonstrating the coverage of environmental assumptions by the Non-IT security objectives. The following table shows the assumption that each Non-IT security objective addresses.

Table 8-2 Non-IT Security Objectives Rationale Mapping

Non-IT Security Objectives	Environmental Assumptions
O.INSTALL	A.MANAGE A.NO_EVIL_ADM A.PEER
O.PHYSICAL	A.LOCATE A.PROTECT A.CONNECT
O.CREDEN	A.COOP

O.INSTALL – By ensuring that the TOE is delivered, installed, managed, and operated in a secure manner, the assumptions A. MANAGE, A.NO_EVIL_ADM, and A.PEER are addressed. This objective ensures that the TOE is managed and administered in a secure manner by a competent and security aware individual in accordance with the administrator documentation.

O.PHYSICAL – By ensuring that the responsible individuals ensure that the TOE is protected from physical attack, the assumptions A.LOCATE, A.PROTECT, and A.CONNECT are addressed because the objective ensures that the TOE is protected from unauthorized physical access.

O.CREDEN – By ensuring that access credentials are adequately protected addresses the assumption A.COOP because it ensures that only those users that are authorized are allowed to gain access to the TOE which supports a benign environment and cooperative users.

Of the definition of the environment in this ST (assumptions, policies, and threats), the assumptions are the only aspects of the environment definition that are Non-IT related. All of the policies and threats are addressed by the IT security objectives. For each assumption, the associated Non-IT Security Objectives there are appropriateness to address the assumptions associated with them in Table 8.2. Given that the Non-IT Security Objectives are met, the assumptions will be achieved.

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 security objectives are completely met by the security functional requirements.

8.2.1 Security Functional Requirements Rationale

The following table provides the correspondence mapping between security objectives for the TOE and the requirements that satisfy them.

Table 8-3 Requirement to Security Objective Correspondence

Requirement	O.AUTHORIZATION	O.DISCRETIONARY_ACCESS	O.AUDITING	O.AUDIT_PROTECTION	O.RESIDUAL_INFORMATION	O.MANAGE	O.ENFORCEMENT	O.PROTECT	O.TRUSTED_PATH	O.LEGAL_WARNING	O.LIMIT_AUTHORIZATION	O.IPSEC	O.ENCRYPTED_DATA	O.ASSURANCE	O.MEDIATE
FAU_GEN.1			X												
FAU_GEN.2			X												
FAU_SAR.1			X			X									
FAU_SAR.2			X												
FAU_SAR.3(a), (b)			X			X									
FAU_SEL.1			X												
FAU_STG.1			X	X											
FAU_STG.3			X			X									
FAU_STG.4			X	X		X									
FCS_COP.1(a) thru (j)												X	X		
FCS_CKM.1(a) thru (e)												X	X		
FCS_CKM.2												X	X		
FCS_CKM.4												X	X		
FCS_CKM_EX.1												X	X		
FCS_CKM_EX.2												X	X		
FDP_ACC.2(a)		X													
FDP_ACC.2(b)		X													
FDP_ACC.2(c)		X													

Requirement	O.AUTHORIZATION	O.DISCRETIONARY_ACCESS	O.AUDITING	O.AUDIT_PROTECTION	O.RESIDUAL_INFORMATION	O.MANAGE	O.ENFORCEMENT	O.PROTECT	O.TRUSTED_PATH	O.LEGAL_WARNING	O.LIMIT_AUTHORIZATION	O.IPSEC	O.ENCRYPTED_DATA	O.ASSURANCE	O.MEDIATE
FDP_ACF.1(a)		X													
FDP_ACF.1(b)		X													
FDP_ACF.1(c)		X													
FDP_IFC.1(a)												X			X
FDP_IFC.1(b)												X			X
FDP_IFF.1(a)												X			X
FDP_IFF.1(b)												X			X
FDP_ITT.1												X			
FDP_RIP.2					X										
FDP_UCT.1								X							
FDP_UIT.1								X							
Note1_EX					X										
FIA_AFL.1	X														
FIA_ATD.1	X	X									X				
FIA_SOS.1	X														
FIA_UAU.1	X														
FIA_UAU.6	X														
FIA_UAU.7	X														
FIA_UID.1	X														
FIA_USB.1_EX		X	X												
FMT_MSA.1(a)		X				X									
FMT_MSA.1(b)		X				X									
FMT_MSA.1(c)						X						X			
FMT_MSA.1(d)						X						X			
FMT_MSA.1(e)		X				X									
FMT_MSA.1(f)		X				X									
FMT_MSA_EX.2								X							
FMT_MSA.3(a)		X				X									
FMT_MSA.3(b)						X						X			
FMT_MSA.3(c)						X						X			
FMT_MSA.3(d)		X				X									
FMT_MSA.3(e)		X				X									
FMT_MTD.1(a)			X			X									
FMT_MTD.1(b)			X			X									
FMT_MTD.1(c)						X		X			X				
FMT_MTD.1(d)	X					X									
FMT_MTD.1(e)	X					X									
FMT_MTD.1(f)	X					X									
FMT_MTD.1(g)			X			X									
FMT_MTD.1(h)						X									
FMT_MTD.1(i)						X				X					
FMT_MTD.1(j)			X			X									
FMT_MTD.1(k)	X														

Requirement	O.AUTHORIZATION	O.DISCRETIONARY_ACCESS	O.AUDITING	O.AUDIT_PROTECTION	O.RESIDUAL_INFORMATION	O.MANAGE	O.ENFORCEMENT	O.PROTECT	O.TRUSTED_PATH	O.LEGAL_WARNING	O.LIMIT_AUTHORIZATION	O.IPSEC	O.ENCRYPTED_DATA	O.ASSURANCE	O.MEDIATE
FMT_MTD.1(l)						X									
FMT_MTD.1(m)						X		X							
FMT_MTD.1(n)						X									
FMT_MTD.1(o)						X									
FMT_MTD.2	X					X									
FMT_MOF.1(a)						X									
FMT_MOF.1(b)						X						X			
FMT_MOF.1(c)	X					X									
FMT_MOF.1(d)						X									
FMT_REV.1(a)						X					X				
FMT_REV.1(b)		X													
FMT_SAE.1	X					X									
FMT_SMF.1						X									
FMT_SMR.1						X					X				
FMT_SMR.3						X									
TRANSFER_PROT_E X.1								X				X			
TRANSFER_PROT_E X.3												X			
FPT_AMT.1							X								
FPT_RPL_EX.1												X			
FPT_RVM.1							X								X
FPT_SEP.1							X	X							
FPT_STM.1			X												
FPT_TRC_EX						X									
FRU_RSA.1	X														
FTA_LSA_EX.1	X														
FTA_MCS_EX.1	X														
FTA_SSL.1	X														
FTA_SSL.2	X														
FTA_SSL.3	X														
FTA_TAB.1										X					
FTA_TSE.1	X														
FTP_TRP.1									X						
ACM_AUT.1														X	
ACM_CAP.4														X	
ACM_SCP.2														X	
ADO_DEL.2															
ADO_IGS.1														X	
ADV_FSP.2															
ADV_HLD.2															
ADV_IMP.1															
ADV_LLD.1															
ADV_RCR.1															

Requirement	O.AUTHORIZATION	O.DISCRETIONARY_ACCESS	O.AUDITING	O.AUDIT_PROTECTION	O.RESIDUAL_INFORMATION	O.MANAGE	O.ENFORCEMENT	O.PROTECT	O.TRUSTED_PATH	O.LEGAL_WARNING	O.LIMIT_AUTHORIZATION	O.IPSEC	O.ENCRYPTED_DATA	O.ASSURANCE	O.MEDIATE
ADV_SPM.1															
AGD_ADM.1														X	
AGD_USR.1															
ALC_DVS.1														X	
ALC_FLR.3														X	
ALC_LCD.1														X	
ALC_TAT.1														X	
ATE_COV.2														X	
ATE_DPT.1														X	
ATE_FUN.1														X	
ATE_IND.2														X	
AVA_MSU.2														X	
AVA_SOF.1														X	
AVA_VLA.2														X	

O.AUTHORIZATION:

FIA_ATD.1 and FMT_MTD.1(d) define data to be used for authentication per user and restrict the ability to initialize authentication data to only authorized administrator, and the ability to modify authentication to authorized administrators and authorized users.

FTA_LSA_EX.1 restricts a user’s capabilities based on the ability for them to logon which can be restricted based upon the ability of a user to logon locally to a given system, the time, and the day.

FIA_AFL.1, FMT_MTD.1(e) and FMT_MTD.2 allow the authorized administrator the ability to set thresholds on the amount of attempts to logon that can be made before a user is locked out and the duration the account locked out.

FIA_SOS.1 defines a metric the authentication mechanism must meet.

FIA_UAU.1, FIA_UID.1 and FIA_UAU.7 require a user to be identified and authenticated before any other TSF-mediation action on their behalf, with the exception of web server access, is allowed and prevent the user requesting access from receiving insightful authentication feedback during the authentication.

FIA_UAU.6 requires a user to be authenticated prior to changing their password.

FTA_SSL.1, FTA_SSL.2, FTA_SSL.3, FMT_MOF.1(c), FMT_MTD.1(k) allow for the authorized user to define and modify a period of user inactivity before the session is locked and for the authorized user or authorized administrator to unlock a locked session as well as initiate the locking of a session. Unlocking a session by an authorized user requires re-authentication.

FMT_MTD.1(f), FTA_TSE.1, and FMT_SAE.1 provide the administrator with the ability to define authentication parameters that further restrict the authentication mechanism which provides access to the TOE.

FTA_MSC.1 allows the administrator to set, and the TSF to enforce, a maximum number of concurrent interactive sessions per user which further restricts access to the TOE.

They allow the authorized administrator the ability to modify the minimum password length and set an expiration limit on authentication data that upon the expiration time the user is prevented from logging on.

FRU_RSA.1 limits access to NTFS volume resources based on quotas, thereby, supporting the ability of the TOE to restrict access to its resources and ensuring that only users that have not exceeded their quota can access NTFS volume resources.

These requirements together restrict access to the TOE by enforcing authentication and identification of users based on the user accounts including user attributes and limits defined by the authorized administrator.

O.DISCRETIONARY_ACCESS:

FDP_ACC.2(a) and FDP_ACF.1(a); FDP_ACC.2(b) and FDP_ACF.1(b); FDP_ACC.2(c) and FDP_ACF.1(c); define several discretionary Security Functional Policies (SFPs), each identifies the subjects and objects which the policy covers, the security attributes that access to objects is based upon, and the rules of access between subjects and objects. The discretionary SFPs allows for the control of access to resources based on the user identity.

FIA_ATD.1 and FIA_USB.1_EX define the security attributes associated with users that used to enforce the SFPs.

FMT_MSA.1(a), FMT_MSA.1(b), FMT_MSA.1(e), FMT_MSA.1(f), FMT_MSA.3(a), FMT_MSA.3(d), FMT_MSA.3(e), and FMT_REV.1(b) restrict the ability to modify object security attributes to authorized users, ensures that the default values are known (permissive or restrictive) for the security attributes used to enforce the SFPs, and ensures that only authorized users can revoke the security attributes used to enforce the SFPs.

These requirements together allow the users the ability to specify, modify, and revoke how objects they are authorized to control can be shared; ensures that the system enforces the sharing specified; and that the security attributes of the users cannot be modified by other than the authorized administrator.

Each of the above requirements together ensure that access is controlled to resources based on user identity and allow authorized users to specify which resources may be accessed by which users.

O.AUDITING:

FAU_GEN.1, FAU_GEN.2, FIA_USB.1_EX, FPT_STM.1, and FMT_MTD.1(g) define the events that must be auditable and ensures that each event shall identify the user that caused the event and the time the event occurred.

FAU_SAR.1, FAU_SAR.2, FAU_SAR.3(a), FAU_SAR.3(b), FAU_STG.1, FAU_STG.3, FAU_STG.4, FMT_MTD.1(j), FMT_MTD.1(a), and FMT_MTD.1(b) ensure that the audit trail is complete and that audit events can be selected and reviewed by only the authorized administrator, and that the audit log (security log) can be managed appropriately by the authorized administrator. Additionally, FAU_SEL.1 provides the capability to the authorized administrator to select the events that will be audited based upon specific attributes (pre-selection of audit events).

Each of the above requirements together ensure the generation of audit records, the adequacy of the content of audit records, and that the audit records are available to and managed by the authorized administrator.

O.AUDIT_PROTECTION:

FAU_STG.1 and FAU_STG.4 require the TOE to restrict access to the audit trail and to prevent the loss of audit data.

By restricting access to the audit trail and preventing the loss of audit data the requirements together ensures the protection of audit records.

O.RESIDUAL_INFORMATION:

FDP_RIP.2 and Note1_EX require the TSF to purge residual data associated with objects and subjects prior to reuse.

Each of the above requirements together ensure that residual data associated with objects and subjects are purged, thereby ensuring that information contained in protected resources does not remain available when the resource is recycled.

O.MANAGE :

FAU_SAR.1, FAU_SAR.3(a), FAU_SAR.3(b), FAU_STG.3, FAU_STG.4, FMT_MTD.1(a), FMT_MTD.1(b), and FMT_MTD.1(j) ensure the authorized administrator can manage audit records.

FMT_MSA.1(a), FMT_MSA.1(b), FMT_MSA.1(e), FMT_MSA.1(f), FMT_MSA.1(c), FMT_MSA.1(d), FMT_MSA.3(a), FMT_MSA.3(b), FMT_MSA.3(c), FMT_MSA.3(d), FMT_MSA.3(e), FMT_MTD.1(c) and FMT_REV.1(a) ensure the authorized administrator can manage attributes used to enforce the SFPs.

FMT_MTD.1(d), FMT_MTD.1(e), FMT_MTD.1(f), FMT_MTD.1(i), FMT_MTD.2, FMT_MOF.1(c), and FMT_SAE.1 ensure the authorized administrator can manage authentication related data. FMT_MTD.1(l), FMT_MTD.1(g), FMT_MTD.1(h), FMT_MTD.1(n) restrict the ability to modify TSF data (including the password complexity requirements). FMT_MTD.1(m) prevents all users (including the authorized administrator) from reading passwords. FMT_MTD.1(o) restricts the initialization of the user security attribute private/public key pair to authorized users and the authorized administrator.

FMT_SMR.1, and FMT_SMR.3 ensure the role of the authorized administrator is enforced.

FMT_SMF.1 ensures the authorized administrator is provided the capability to change and maintain security relevant data (e.g. audit policy, account policy, etc).

FMT_MOF.1(a) and FMT_MOF.1(b) ensure the authorized administrator can manage the audit function and the function to protect TSF data during transmission. FMT_MOF.1(d) ensure the authorized administrator can manage the web server policy that controls the access to web server content.

FPT_TRC_EX ensures that TSF data can be replicated between parts of the TOE to enable TSFs to have the most recent TSF data.

Together the above requirements ensure that the administrator can manage data (audit records, attributes used to enforce the SFPs, authentication data), manage functions (audit, protection of data in transmission, replication of TSF data), and ensure that the authorized user and administrator roles are enforced. Changes to specific TSF data are distributed throughout the TOE assisting in the management of a distributed TOE.

Each of the above requirements contributes to and together ensures that the authorized administrator can manage the TOE securely.

O.ENFORCEMENT:

FPT_RVM.1, FPT.SEP.1, and FPT_AMT.1 ensure the TOE makes and enforces the decisions of the TSPs and that the TSF is protected from interference that would prevent it from performing its functions.

Together the above requirements ensure that the underlying abstract machine relied upon by the TSF is operating correctly, and that the TSF continues to operate effectively to uphold the TSPs.

Each of the above requirements together ensures that the organizational policies are enforced.

O.PROTECT:

FMT_MTD.1(c) ensures that user security attributes which the SFPs are based upon can only be initialized and modified by an authorized administrator. FMT_MSA_EX.2 ensures that only valid password values are accepted by the TOE as security attributes supporting the ability for the TOE to protect itself.

FMT_MTD.1(m) protects the TOE authentication data by preventing authentication from being read by any user (including the administrator).

TRANSFER_PROT_EX and FPT_SEP.1 ensure that the TOE provides TSF protection of system resources and maintains a separate domain for the TSF.

FDP_UCT.1 and FDP_UIT.1 ensure that the data communication between web users and the web server is protected from unauthorized disclosure and modification.

Together the requirements ensure that the TSF data is protected from modification, protected in transmission, and that the TSF cannot be modified in an unauthorized manner.

Each of the above requirements contributes to and together ensures that a separate domain is maintained for the TSF and the TSF protects its own data and resources.

O.TRUSTED_PATH:

FTP_TRP.1 ensures the TOE includes a capability for the user to utilize a trusted path with the TSF for initial logon and session unlocking.

The above requirement ensures there is a mechanism that allows the user to assuredly communicate with the TSF, and not another entity pretending to be the TSF, during initial user authentication

O.LEGAL_WARNING:

FTA_TAB.1 requires the TOE to provide the capability of displaying a banner before login.

FMT_MTD.1(i) restricts the modification of the banner content to an authorized administrator.

Each of the above requirements together ensure that a banner can be displayed before login containing a warning defined by an authorized administrator to advise users of legal issues involving the misuse of the TOE before access to resources is allowed.

O.LIMIT_AUTHORIZATION:

FMT_SMR.1; FIA_ATD.1; FMT_MTD.1(c); and FMT_REV.1(a) require the TOE to provide the capability to limit user authorizations by the definition of roles, the user privileges, and the revocation of security-relevant authorizations.

By ensuring that security attributes associated with users can only be assigned and revoked by the administrator and that the security attributes allow for specific roles to be enforced, these requirements ensure that the capabilities of users can be limited.

Each of the above requirements together ensures the capability to limit the extent of each user's authorizations.

O.IPSEC:

FDP_ITT.1, TRANSFER_PROT_EX.1, FMT_MOF.1(b) ensures the capability to protect TSF data from disclosure and modification when in transmission between distributed parts of the TOE and provide management support for these functions.

FPT_RPL_EX.1 and TRANSFER_PROT_EX.3 ensures data in transmission is protected by rejecting or auditing TSF data for which a replay of TSF data is detected.

FDP_IFC.1(a) and FDP_IFF.1(a) ensure that the IPSec filters can be used to control the flow of traffic amongst the different systems (or TSFs) within the TOE. FDP_IFC.1(b) and FDP_IFF.1(b) ensure that the TOE may be configured to prevent unsolicited traffic into the TSF.

FMT_MSA.1(c), FMT_MSA.1(d), FMT_MSA.3(b), and FMT_MSA.3(c) restrict the ability to modify security attributes to authorized users and ensures that known default values are defined for the security attributes used to enforce the SFP.

Additionally, all the cryptographic requirements (all FCS_COP and FCS_CKM related requirements) support IPSec in its application of security services that involve digital signatures, encryption, decryption, the hashing, and other services. These services support of protection and control of traffic in transmission between physically separate parts of the TOE.

The above requirements together protect the authorized administrator with the capability to configure the system to protect system data in transmission between distributed parts of the TOE.

O.ENCRYPTED_DATA:

FCS_COP.1(a) thru (j), FCS_CKM.1(a) thru (e), FCS_CKM.2, FCM_CKM.4, FCS_CKM_EX.1 and FCS_CKM_EX.2 prevents the decryption of encrypted data if the user attempting decryption is not the user that encrypted the data and supports cryptographic operations that support the encryption and decryption of data such as hashing, key generation, key validation, key handling and key storage.

These requirements together prevent users from decrypting data they did not encrypt and ensures that only those users that encrypted data can decrypt that data.

O.ASSURANCE:

AGD_ADM.1, AVA_MSU.2, ADO_IGS.1 support that the TOE is installed and configured properly. ACM_AUT.1, ACM_CAP.4, ACM_SCP.2, ALC_DVS.1, ALC_FLR.3, ALC_LCD.1, and ALC_TAT.1 support that the TOE is protected during its development. ATE_COV.2, ATE_DPT.1, ATE_FUN.1, ATE_IND.2, and AVA_VLA.2 support that the TOE is sufficiently protect and can protect itself against the casual attacker.

O.MEDIATE:

FDP_IFC.1(a), FDP_IFC.1(b), FDP_IFF.1(a), and FDP_IFF.1(b) support that the TOE ensures all network packets that flow through the TOE are subject to information flow policies. FPT_RVM.1 ensures the policy cannot be bypassed.

8.2.2 SAR Rationale

This ST contains the assurance requirements from the CC EAL4 assurance package augmented with ALC_FLR.3. The CC allows assurance packages to be augmented, which allows the addition of assurance components from the CC not already included in the EAL. Augmentation was chosen to provide the added assurance that is provided by defining flaw remediation procedures and correcting security flaws (ALC_FLR.3). This ST is based on good rigorous commercial development practices and has been developed for a generalized environment for a TOE that is generally available and does not require modification to meet the security needs of the environment specified in this ST.

The EAL chosen is based on the statement of the security environment (threats, organizational policies, assumptions) and the security objectives defined in this ST. The sufficiency of the EAL chosen (EAL4 augmented) is justified based on those aspects of the environment that have impact upon the assurance needed in the TOE. Users will act in a cooperative manner in a benign environment (A.COOP, O.CREDEN); the administrative staff is conscientious and not hostile (A.NO_EVIL_ADM); the TOE is designed and implemented in a manner which ensures the security policies are enforced (O.ENFORCEMENT); and, the TOE is physically protected (O.PHYSICAL) and properly and securely configured (O.INSTALL). Given these aspects, a TOE based on good commercial development practices is sufficient. The CC states that EAL 4 permits a developer to gain the maximum assurance from positive security engineering based on good commercial development practices which, though rigorous, do not require substantial specialist knowledge, skills, and other resources. Given the amount of assurance deemed necessary to meet the security environment and objectives of the TOE and the intent of EAL 4, EAL 4 is an appropriate level of assurance for the TOE described in this ST. Thereby, EAL4 augmented is an appropriate level of assurance for the TOE.

While the EAL chosen is not the same as is specified in the CAPP, this ST remains CAPP conformant because the EAL chosen in this ST (EAL4 augmented) is hierarchical to the EAL specified in the CAPP (EAL3). EAL 4 augmented was chosen instead of EAL 3 because the ST authors chose to achieve the highest level of assurance feasible based on current development practices.

8.2.3 Requirement Dependency Rationale

Table 8-4 depicts the satisfaction of all functional requirement dependencies. For each functional requirement included in the ST, the CC dependencies are identified in the column "Dependencies." Additionally, all operations performed upon requirements were reviewed. None were found to add any dependencies in addition to those identified in the CC.

For explicitly stated requirements (those ending with “_EX”), the CC dependencies identified for similar requirements were used as guidance to identify their dependencies, and additionally, all the explicitly included requirements in the ST were considered. The following pertains:

- For FIA_USB.1_EX and Note1_EX, there is no change in the dependencies from the CC identified dependencies for the CC requirements these explicit requirements are based upon (FIA_USB.1 and FDP_RIP.2) considering the changes between the CC requirements and the explicit requirements.
- For TRANSFER_PROT_EX.1, there are no CC identified dependencies for the CC requirements this explicit requirement is based upon (FPT_ITT.1). Considering the changes between the CC requirements and the explicit requirements, the TRANSFER_PROT_EX explicit requirements is dependent upon the TOE providing the functionality to allow the administrator to enable or disable the functionality described in these explicit requirements and the TOE providing cryptographic functionality. Therefore, TRANSFER_PROT_EX is dependent upon FMT_MOF.1(b) and FCS_COP.1.
- For TRANSFER_PROT_EX.3, the dependency is to an explicit requirement (TRANSFER_PROT_EX.1) which is similar to the CC identified dependency and acceptable considering the difference between the explicit requirements and the similar CC requirements (TRANSFER_PROT_EX.1 and FPT_ITT.1; and TRANSFER_PROT_EX.3 and FPT_ITT.3). However, the explicit requirement requires that the TOE provides cryptographic functionality and is, therefore, dependent upon FCS_COP.1.
- For FPT_TRC_EX, the dependency is to an explicit requirement (TRANSFER_PROT_EX) which is similar to the CC identified dependency and acceptable considering the difference between the explicit requirements and the similar CC requirements (TRANSFER_PROT_EX.1 and FPT_ITT.1; and FPT_TRC_EX and FPT_TRC.1).
- For FCS_CKM_EX.1 and FCS_CKM_EX.2, there are no CC requirements that address the nature of these requirements.
- For FPT_RPL_EX.1, there are no CC identified dependencies for the CC requirement this explicit requirement is based upon (FPT_RPL.1). However, the explicit requirement requires that the TOE provides cryptographic functionality and is, therefore, dependent upon FCS_COP.1.
- For FMT_MSA_EX.2, the dependencies identified are the same dependencies as the CC identified dependencies for the CC requirement, FMT_MSA.2, this explicit requirement is based upon.
- For the FCS_COP.1 requirement, the CC identifies the following dependency: FDP_ITC.1 or FCS_CKM.1, FCS_CKM.4, and FMT_MSA.2. The following dependency for this requirement is not applicable and the rationale is as follows:
 - FDP_ITC.1: this requirement applies to user data that is imported from outside of the TSF Scope of Control (TSC) and concerned with applying rules to the imported data (e.g. ignore security attributes associated with data when imported). There is no user data within the TOE that is imported from outside the TSC and, therefore, this requirement is not applicable.
 - FMT_MSA.2: this requirement is concerned with ensuring that only secure values are accepted for security attributes. There are no security attributes entered by users within the context of the operations specified by FCS_COP.1, therefore, FMT_MSA.2 is not applicable to FCS_COP.1.

The component number in column “Satisfied Component No” denotes the requirement(s) that is included in this ST to meet the dependencies of each functional requirement. The component number used in the

column “Satisfied Component No.” is the component number used to identify each ST Functional Requirement in column “Component No.” With the exception of the requirement for which a rationale is provided above (FCS_COP.1), all the dependencies are satisfied by component numbers of requirements included in this ST. Therefore, all dependencies have been satisfied.

Note that the letters “a” through “k” are used to enumerate iterations of the requirements in the column “ST Functional Requirement.”

Table 8-4 Dependency Rationale Mapping

Component No.	ST Functional Requirement	Dependencies	Satisfied Component No.
1.	FAU_GEN.1	FPT_STM.1	83
2.	FAU_GEN.2	FAU_GEN.1 FIA_UID.1	1, 38
3.	FAU_SAR.1	FAU_GEN.1	1
4.	FAU_SAR.2	FAU_SAR.1	3
5.	FAU_SAR.3(a), (b)	FAU_SAR.1	3
6.	FAU_SEL.1	FAU_GEN.1, FMT_MTD.1	1, 56
7.	FAU_STG.1	FAU_GEN.1	1
8.	FAU_STG.3	FAU_STG.1	7
9.	FAU_STG.4	FAU_STG.1	7
10.	FCS_COP.1 (a) –(j)	FDP_ITC.1 or FCS_CKM.1 FCS_CKM.4 FMT_MSA.2	12, 14, 50
11.	FCS_CKM.1(a) –(e)	FCS_CKM.2 or FCS_COP.1 FCS_CKM.4, FMT_MSA.2	11, 14, 50
12.	FCS_CKM.2	FDP_ITC.1 or FCS_CKM.1 FCS_CKM.4 FMT_MSA.2	12, 14, 50
13.	FCS_CKM.4	FDP_ITC.1 or FCS_CKM.1 FMT_MSA.2	12, 50
14.	FCS_CKM_EX.1	None	
15.	FCS_CKM_EX.2	None	
16.	FDP_ACC.2(a)	FDP_ACF.1	20
17.	FDP_ACC.2(b)	FDP_ACF.1	21
18.	FDP_ACC.2(c)	FDP_ACF.1	22
19.	FDP_ACF.1(a)	FDP_ACC.1 FMT_MSA.3	17, 51

Component No.	ST Functional Requirement	Dependencies	Satisfied Component No.
20.	FDP_ACF.1(b)	FDP_ACC.1 FMT_MSA.3	18, 54
21.	FDP_ACF.1(c)	FDP_ACC.1 FMT_MSA.3	19, 55
22.	FDP_IFC.1(a)	FDP_IFF.1	20
23.	FDP_IFC.1(b)	FDP_IFF.1	21
24.	FDP_IFF.1(a)	FDP_IFC.1 FDP_MSA.3	23, 52
25.	FDP_IFF.1(b)	FDP_IFC.1 FDP_MSA.3	24, 53
26.	FDP_ITT.1	FDP_ACC.1 or FDP_IFC.1	17
27.	FDP_RIP.2	None	
28.	FDP_UCT.1	[FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path] [FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control]	92, 18
29.	FDP_UIT.1	[FDP_ACC.1 Subset access control, or FDP_IFC.1 Subset information flow control] [FTP_ITC.1 Inter-TSF trusted channel, or FTP_TRP.1 Trusted path]	92, 18
30.	Note1_EX	None	
31.	FIA_AFL.1	FIA_UAU.1	35
32.	FIA_ATD.1	None	
33.	FIA_SOS.1	None	
34.	FIA_UAU.1	FIA_UID.1	38
35.	FIA_UAU.6	None	
36.	FIA_UAU.7	FIA_UAU.1	35
37.	FIA_UID.1	None	
38.	FIA_USB.1_EX	FIA_ATD.1	33
39.	FMT_MOF.1(a)	FMT_SMR.1 FMT_SMF.1	62, 63
40.	FMT_MOF.1(b)	FMT_SMR.1 FMT_SMF.1	75, 74
41.	FMT_MOF.1(c)	FMT_SMR.1 FMT_SMF.1	73, 74

Component No.	ST Functional Requirement	Dependencies	Satisfied Component No.
42.	FMT_MOF.1(d)	FMT_SMR.1 FMT_SMF.1	73, 74
43.	FMT_MSA.1(a)	FDP_ACC.1 or FDP_IFC.1, FMT_SMR.1 FMT_SMF.1	17, 73, 74
44.	FMT_MSA.1(b)	FDP_ACC.1 or FDP_IFC.1, FMT_SMR.1 FMT_SMF.1	17, 73, 74
45.	FMT_MSA.1(c)	FDP_ACC.1 or FDP_IFC.1, FMT_SMR.1 FMT_SMF.1	23, 73, 74
46.	FMT_MSA.1(d)	FDP_ACC.1 or FDP_IFC.1, FMT_SMR.1 FMT_SMF.1	24, 73, 74
47.	FMT_MSA.1(e)	FDP_ACC.1 or FDP_IFC.1, FMT_SMR.1 FMT_SMF.1	18, 73, 74
48.	FMT_MSA.1(f)	FDP_ACC.1 or FDP_IFC.1, FMT_SMR.1 FMT_SMF.1	19, 73, 74
49.	FMT_MSA_EX.2	ADV_SPM.1 FDP_ACC.1 or FDP_IFC.1 FMT_MSA.1 FMT_SMR.1	EAL4, 17, 44, 75
50.	FMT_MSA.3(a)	FMT_MSA.1(a) FMT_SMR.1	44, 75
51.	FMT_MSA.3(b)	FMT_MSA.1(c) FMT_SMR.1	46, 75
52.	FMT_MSA.3(c)	FMT_MSA.1(d) FMT_SMR.1	46, 75
53.	FMT_MSA.3(d)	FMT_MSA.1(e) FMT_SMR.1	48, 75
54.	FMT_MSA.3(e)	FMT_MSA.1(f) FMT_SMR.1	49, 75

Component No.	ST Functional Requirement	Dependencies	Satisfied Component No.
55.	FMT_MTD.1(a) – (g)	FMT_SMR.1 FMT_SMF.1	73, 74
56.	FMT_MTD.1(b)	FMT_SMR.1 FMT_SMF.1	73, 74
57.	FMT_MTD.1(c)	FMT_SMR.1 FMT_SMF.1	73, 74
58.	FMT_MTD.1(d)	FMT_SMR.1 FMT_SMF.1	73, 74
59.	FMT_MTD.1(e)	FMT_SMR.1 FMT_SMF.1	73, 74
60.	FMT_MTD.1(f)	FMT_SMR.1 FMT_SMF.1	73, 74
61.	FMT_MTD.1(g)	FMT_SMR.1 FMT_SMF.1	73, 74
62.	FMT_MTD.1(h)	FMT_SMR.1 FMT_SMF.1	73, 74
63.	FMT_MTD.1(i)	FMT_SMR.1 FMT_SMF.1	73, 74
64.	FMT_MTD.1(j)	FMT_SMR.1 FMT_SMF.1	73, 74
65.	FMT_MTD.1(k)	FMT_SMR.1 FMT_SMF.1	73, 74
66.	FMT_MTD.1(l)	FMT_SMR.1 FMT_SMF.1	73, 74
67.	FMT_MTD.1(m)	FMT_SMR.1 FMT_SMF.1	73, 74
68.	FMT_MTD.1(n), (o)	FMT_SMR.1 FMT_SMF.1	74, 73
69.	FMT_MTD.2	FMT_MTD.1(e) FMT_SMR.1	59, 74
70.	FMT_REV.1(a)	FMT_SMR.1	75
71.	FMT_REV.1(b)	FMT_SMR.1	75
72.	FMT_SAE.1	FMT_SMR.1 FPT STM.1	75, 83
73.	FMT_SMF.1	None	

Component No.	ST Functional Requirement	Dependencies	Satisfied Component No.
74.	FMT_SMR.1	FIA_UID.1	38
75.	FMT_SMR.3	FMT_SMR.1	75
76.	TRANSFER_PROT_EX.1	FMT_MOF.1(b) FCS_COP.1	41 11
77.	TRANSFER_PROT_EX.3	FPT_ITT.1 (equivalent to explicit requirement TRANSFER_PROT_EX.1) FCS_COP.1	77 11
78.	FPT_RPL_EX.1	FCS_COP.1	11
79.	FPT_TRC_EX.1	TRANSFER_PROT_EX.1	77
80.	FPT_RVM.1	None	
81.	FPT_SEP.1	None	
82.	FPT_STM.1	None	
83.	FRU_RSA.1	None	
84.	FTA_LSA_EX.1	None	
85.	FTA_MCS_EX.1	FIA_UID.1 FMT_SMF.1	38 74
86.	FTA_SSL.1	FIA_UAU.1	35
87.	FTA_SSL.2	FIA_UAU.1	35
88.	FTA_SSL.2	FIA_UAU.1	35
89.	FTA_TAB.1	FMT_MTD.1.1(i)	64
90.	FTA_TSE.1	None	
91.	FTP_TRP.1	None	

8.2.4 Explicitly Stated Requirements Rationale

The ST includes the following explicitly stated requirements: FCS_CKM_EX.1; FCS_CKM_EX.2; Note1_EX; FIA_USB.1_EX; FMT_MSA_EX.2; TRANSFER_PROT_EX; FPT_RPL_EX; FPT_TRC_EX.1; Note1_EX; and, FIA_USB.1_EX, referred to as FDP_RIP.2.Note1 and FIA_USB.1 in the CAPP, are included in the CAPP along with a rationale for each requirement.

FCS_CKM_EX.1: The CC cryptographic support section does not specifically address the concepts of key validation techniques. Although closely tied to generated keys, these concepts typically get implemented just after, not during, the actual generation of a key. In this ST, FCS_CKM_EXP.1 allows for specifically addressing these key management-related concepts in support of O.IPSEC and O.ENCRYPTED_DATA.

FCS_CKM_EX.2: The CC does not provide components for key handling and storage. In this ST, FCS_CKM_EX.2 allows for specifically addressing these key management-related concepts in support of O.IPSEC and O.ENCRYPTED_DATA.

FMT_MSA_EX.2: To address the TOE functionality of the ability to enforce that passwords meet the password complexity requirements in support of the objective O.PROTECT, the FMT_MSA.2 CC

requirement was considered. However, FMT_MSA.2 enforces that all security attributes are “secure” while the TOE functionality is more adequately expressed as ensuring the password security attributes are “valid” in that they meet the password complexity requirements defined by the administrator.

TRANSFER_PROT_EX.1: To address the TOE functionality of the protection of data in transmission between different parts of the TOE to support the objective O.PROTECT, the FPT_ITT.1 CC requirement was considered. However, because FPT_ITT.1 prescribes functionality beyond what is required to meet O.PROTECT and O.IPSEC, the ST authors created the explicit requirement TRANSFER_PROT_EX. The functionality to “always” protect data in transmission between separate parts of the TOE is not necessary to meet the objective O.PROTECT (to protect TSF data) because of the physical protection of all parts of the TOE as required by the Non-IT security objective O.PHYSICAL. The ST authors added the words “be able to” to the requirement to provide the desired flexibility in the evaluated configuration to meet the objectives O.PROTECT and O.IPSEC. The ST authors also qualified the requirement to apply to protecting all data instead of only TSF data. This change allows the authorized administrator to be able to disable this functionality and remain within the evaluated configuration.

TRANSFER_PROT_EX.3: To address the TOE functionality of the protection of data in transmission between different parts of the TOE to support the objective O.IPSEC, the FPT_ITT.3 CC requirement was considered. However, because FPT_ITT.3 prescribes functionality beyond what is required to meet O.IPSEC, the ST authors created the explicit requirement TRANSFER_PROT_EX.3. The ST authors qualified the requirement to apply only to protecting all data instead of only TSF data.

FPT_RPL_EX: To address the TOE functionality to detect replay in support of O.IPSEC the CC requirement FPT_RPL.1 was considered. However, because FPT_RPL prescribes functionality beyond what is needed to support O.IPSEC, in that the functionality required must always be enforced. The ST authors created the explicit requirement FPT_RPL_EX which mandates the ability to detect replays, however, this functionality need not always be configured to be enforced in its evaluated configuration to do so (similar to FAU_GEN.1).

FPT_TRC_EX: To address the TOE functionality of TSF data replication to support the objective O.MANAGE, the FPT_TRC.1 CC requirement was considered. However, because FPT_TRC.1 prescribes functionality beyond what is required to meet O.MANAGE which the TOE does not implement, the ST authors created the explicit requirement FPT_TRC_EX. Ensuring that data is “totally” consistent between separate TSFs in a distributed TOE appears to be the intent of FPT_TRC.1, which is not required by any TOE Objectives. The ST authors chose to create an explicit requirement, FPT_TRC_EX, to ensure that TSF data changed at one TSF is copied to other TSFs and that the target TSF will only accept the changed TSF data if it is more recent than the local copy of that TSF data. FPT_TRC_EX supports the TOE objective O.MANAGE by ensuring that changes to important TSF data are copied to support the accuracy and enforcement of TSF data at each TSF.

FTA_MCS_EX: FTA_MCS_EX is based upon the FTA_MCS.1 CC requirement and is written explicitly because this functionality is enforced only on members of a domain. Additionally, FTA_MCS_EX replaces the assignment in FTA_MCS.1, which allows the ST author to enter a default amount of concurrent sessions allowed, with the ability for the authorized administrator to set this limit. This change introduces a dependency upon FMT_SMF.1 which is addressed in item i (initialize and modify user security attributes).

FTA_LSA_EX: FTA_LSA_EX is based upon the FTA_LSA.1 CC requirement and is written explicitly because only this requirement only applies to members of a domain.

The assurance requirements are still applicable and appropriate with the inclusion of these explicitly stated requirements. The explicitly stated requirements do not demand any additional documentary evidence other than what is required at EAL4.

8.2.5 Internal Consistency and Mutually Supportive Rationale

The selected requirements are internally consistent and fully compliant with the CAPP. The ST includes all of the functional requirements from the CAPP and additional requirements to reflect additional functionality, compatible with the CAPP requirements. All operations that have been performed on the

additional requirements are in accordance with the CC. The ST includes no instance of a requirement that contradicts another requirement in the ST. In instances where different requirements apply to the same events or types of data, the requirements and the operations performed within the requirements do not contradict each other.

The selected requirements together form a mutually supportive whole by the satisfaction of all dependencies as demonstrated in Table 8-4; the mapping and suitability of the requirements to security objectives as justified in Section 8.2.1; the inclusion of architectural requirements FPT_RVM.1 and FPT_SEP.1 to protect the TSF, the inclusion of audit requirements to detect attacks of other security functional requirements; and the inclusion of security management requirements to ensure proper configuration and control of other security functional requirements.

8.2.6 SOF Rationale

The TOE minimum SOF of SOF-medium was chosen to be consistent with the CAPP. The explicit SOF claim for the authentication mechanism described in FIA_SOS.1 and FIA_UAU.1 of guessing a password is stronger than that specified in the CAPP and is in turn consistent with the security objectives described in Section 8.2.1.

The SOF-medium strength level is sufficient to meet the objectives of the TOE given the security environment described in the ST, specifically given the assumption A.COOP (Authorized users possess the necessary authorization to access at least some of the information management by the TOE and are expected to act in a cooperating manner in a benign environment.)

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 and the assurance measures address the assurance measures.

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 8-5. The collection of assurance measures work together to address all of the SARs as indicated in Table 8-6. The SFs and assurance measures described in the TSS and indicated in the tables below are all necessary for the required security functionality in the TSF.

Table 8-5 Requirement to Security Function Correspondence

Requirement	Audit	User Data Protection	Cryptographic Protection	I & A	Security Management	TSF Protection	Resource Utilization	Session Locking
FAU_GEN.1	X							
FAU_GEN.2	X							
FAU_SAR.1	X							
FAU_SAR.2	X							
FAU_SAR.3(a), (b)	X							
FAU_SEL.1	X							

Requirement	Audit	User Data Protection	Cryptographic Protection	I & A	Security Management	TSF Protection	Resource Utilization	Session Locking
FAU_STG.1	X							
FAU_STG.3	X							
FAU_STG.4	X							
FCS_COP.1(a) thru (j)			X					
FCS_CKM.1(a) thru (e)			X					
FCS_CKM.2			X					
FCS_CKM.4			X					
FCS_CKM_EX.1			X					
FCS_CKM_EX.2			X					
FDP_ACC.2(a)		X						
FDP_ACC.2(b)		X						
FDP_ACC.2(c)		X						
FDP_ACF.1(a)		X						
FDP_ACF.1(b)		X						
FDP_ACF.1(c)		X						
FDP_IFC.1(a)		X						
FDP_IFC.1(b)		X						
FDP_IFF.1(a)		X						
FDP_IFF.1(b)		X						
FDP_ITT.1		X						
FDP_RIP.2		X						
FDP_UCT.1		X						
FDP_UIT.1		X						
Note1_EX		X						
FIA_AFL.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_EX				X				
FMT_MSA.1(a)		X						
FMT_MSA.1(b)		X						
FMT_MSA.1(c)		X						
FMT_MSA.1(d)		X						
FMT_MSA.1(e)		X						
FMT_MSA.1(f)		X						
FMT_MSA_EX.2					X			
FMT_MSA.1(d)		X						
FMT_MSA.3 (a)		X						
FMT_MSA.3 (b)		X						
FMT_MSA.3 (c)		X						
FMT_MSA.3 (d)		X						

Requirement	Audit	User Data Protection	Cryptographic Protection	I & A	Security Management	TSF Protection	Resource Utilization	Session Locking
FMT_MSA.3 (e)		X						
FMT_MTD.1(a)	X				X			
FMT_MTD.1(b)					X			
FMT_MTD.1(c)					X			
FMT_MTD.1(d)					X			
FMT_MTD.1(e)					X			
FMT_MTD.1(f)					X			
FMT_MTD.1(g)						X		
FMT_MTD.1(h)					X			
FMT_MTD.1(i)				X				
FMT_MTD.1(j)	X							
FMT_MTD.1(k)								X
FMT_MTD.1(l)					X			
FMT_MTD.1(m)					X			
FMT_MTD.1(n)					X			
FMT_MTD.1(o)					X			
FMT_MTD.2					X			
FMT_MOF.1(a)					X			
FMT_MOF.1(b)					X			
FMT_MOF.1(c)								X
FMT_REV.1(a)					X			
FMT_REV.1(b)		X						
FMT_SAE.1					X			
FMT_SMR.1					X			
FMT_SMR.3				X				
TRANSFER_PROT_EX.1						X		
TRANSFER_PROT_EX.3						X		
FPT_AMT.1						X		
FPT_RPL_EX.1						X		
FPT_RVM.1						X		
FPT_SEP.1						X		
FPT_STM.1						X		
FPT_TRC_EX						X		
FRU_RSA.1							X	
FTA_LSA_EX.1				X				
FTA_MCS_EX.1				X				
FTA_SSL.1								X
FTA_SSL.2								X
FTA_SSL.2								X
FTA_TAB.1				X				
FTA_TSE.1				X				
FTP_TRP.1				X				

Table 8-6 Assurance Requirements to Assurance Measures Mappings

Requirement	Process Assurance	Delivery and Guidance	Design Documentation	Test	Vulnerability Assessment
ACM_AUT.1	X				
ACM_CAP.4	X				
ACM_SCP.2	X				
ADO_DEL.2		X			
ADO_IGS.1		X			
ADV_FSP.2			X		
ADV_HLD.2			X		
ADV_IMP.1			X		
ADV_LLD.1			X		
ADV_RCR.1			X		
ADV_SPM.1			X		
AGD_ADM.1		X			
AGD_USR.1		X			
ALC_DVS.1	X				
ALC_FLR.3	X				
ALC_LCD.1	X				
ALC_TAT.1	X				
ATE_COV.2				X	
ATE_DPT.1				X	
ATE_FUN.1				X	
ATE_IND.2				X	
AVA_MSU.2					X
AVA_SOF.1					X
AVA_VLA.2					X

9. Additional Protection Profile References

This section identifies additional PPs to which conformance is not claimed but were used as a source to identify additional requirements applicable to the TOE. The additional PPs are the PP for Single-level OS' in Environments Requiring Medium Robustness (SLOSPP) and the U.S. Government Web Server Protection Profile (WEB Server PP) for Basic Robustness Environments. A subsection for each of these PPs is included in this section that provides further details regarding how this ST relates to each of the PPs referenced.

9.1 Protection Profile for Single-level Operating Systems (SLOSPP) Reference

This TOE is an OS and while conforming with the CAPP it does provide additional security functionality that meets specific requirements from the SLOSPP. These additional requirements mandate additional security functionality and do not conflict with any CAPP requirements as demonstrated in 7.1.2 CAPP Differences and Enhancements.

The requirements included in this ST that were based upon SLOSPP requirements are:

- FCS_COP.1(a) thru (j) Cryptographic Operation,
- FCS_CKM.1(a) thru FCS_CKM.1(e) Cryptographic Key Generation,
- FCS_CKM.2 Cryptographic Key Distribution,
- FCS_CKM.4 Cryptographic Key Zeroization,
- FCS_CKM_EX.1 Cryptographic Key Validation and Packaging,
- FCS_CKM_EX.2 Cryptographic Key Handling and Storage,
- FDP_ITT.1 Basic Internal Protection,
- FIA_UAU.6 Re-authenticating,
- FMT_MSA_EX.2 Valid Password Security Attributes,
- TRANSFER_PROT_EX.1 Internal TSF Data Transfer Protection,
- FPT_TRC_EX Internal TSF Data Consistency,
- TRANSFER_PROT_EX.3 Internal TSF Data Integrity Monitoring,
- FPT_RPL_EX Replay Detection,
- FTA_LSA_EX.1 Limit on Scope of Selectable Attributes, and
- FTA_MCS_EX.1 Basic Limitation on Multiple Concurrent Sessions.

9.2 Web Server PP Reference

This TOE includes a web server which provides security functionality that meets several requirements U.S. Government WEB Server PP. These additional requirements mandate additional security functionality and do not conflict with any CAPP requirements as demonstrated in 7.1.2 CAPP Differences and Enhancements.

The following requirements included in this ST which are based upon WEB Server PP requirements are:

- FDP_ACC.2(b) WEBUSER Complete Access Control,
- FDP_ACC.2(c) Content-Provider Complete Access Control,

- FDP_ACF.1(b) WEBUSER Access Control Functions,
- FDP_ACF.1(c) Content-Provider Access Control Functions,
- FDP_UCT.1 WEBUSER SFP Basic Data Exchange Confidentiality,
- FDP_UIT.1 WEBUSER SFP Data Exchange Integrity,
- FMT_MOF.1(d) Management of Web Server,
- FMT_MSA.1(e) Management of WEBUSER Object Security Attributes,
- FMT_MSA.1(f) Management of Content-Provider Object Security Attributes,
- FMT_MSA.3(d) WEBUSER Static Attribute Initialization,
- FMT_MSA.3(e) Content-Provider Static Attribute Initialization, and
- FTA_SSL.3 WEBUSER TSF-Initiated Termination.

APPENDIX A—List of Acronyms

3DES	Triple DES
ACE	Access Control Entry
ACL	Access Control List
ACM	Access Control Management
ACP	Access Control Policy
AD	Active Directory
AES	Advanced Encryption Standard
AGD	Administrator Guidance Document
AH	Authentication Header
ANSI	American National Standards Institute
API	Application Programming Interface
CA	Certificate Authority
CALG	Confidentiality Algorithm
CAPP	Controlled Access Protection Profile
CBC	Cipher Block Chaining
CC	Common Criteria
CCSE	Canadian Communication Security Establishment
CD-ROM	Compact Disk Read Only Memory
CI	Configuration Item
CIFS	Common Internet File System
CM	Configuration Management; Control Management
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
DPAPI	Data Protection API
DC	Domain Controller
DES	Data Encryption Standard
DFS	Distributed File System
DH	Diffie-Hellman
DHCP	Dynamic Host Configuration Protocol
DFS	Distributed File System
DNS	Domain Name System
DoS	Denial of Service
DO	Delivery Operation
DS	Directory Service
DSA	Digital Signature Algorithm
EAL	Evaluation Assurance Level
ECB	Electronic Code Book
EFS	Encrypting File System
ESP	Encapsulating Security Protocol
EFW	Enhanced Write Filter
FEK	File Encryption Key
FIPS	Federal Information Processing Standard
FRS	File Replication Service
FSMO	Flexible Single Master Operation
GB	Gigabyte
GC	Global Catalog
GHz	Gigahertz

GINA	Graphical Identification and Authentication
GPC	Group Policy Container
GPO	Group Policy Object
GPT	GUID Partition Table; Group Policy Template
GUI	Graphical User Interface
GUID	Globally Unique Identifiers
HMAC	Hash-Based Message Authentication Code
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
ICS	Internet Connection Sharing
ID	Identification
IEC	International Electro-technical Commission
IETF	Internet Engineering Task Force
IFS	Installable File System
IIS	Internet Information Services
IIS6	IIS Version 6.0
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

ISATAP	Intra-site Automatic Tunnel Addressing Protocol
ISO	International Organization for Standardization
IT	Information Technology
KDC	Key Distributed Center
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LPC	Local Procedure Call
LSA	Local Security Authority
LSASS	LSA Subsystem Service
MAC	Message Authentication Code
MB	Megabyte
MBR	Master Boot Record
MMC	Microsoft Management Console
NAT	Network Address Translation
NIST	National Institute of Standards and Technology
NTFS	New Technology File System
NSA	National Security Agency
NTLM	New Technology LAN Manager
OLE	Object Linking and Embedding
OS	Operating System
PAE	Physical Address Extension
PC/SC	Personal Computer/Smart Card
PDC	Primary DC
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
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
SOF	Strength of Function
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
TSC	TOE Scope of Control
TSF	TOE Security Functions
TSS	TOE Summary Specification
UI	User Interface
UID	User Identifier
UNC	Universal Naming Convention
U.S.	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	WEBUSER
WMD	Windows Driver Model
WMI	Windows Management Instrumentation
WSC	Windows Security Center
WWW	World-Wide Web
X86	Intel Microprocessors

Appendix B—TOE Component Decomposition

Certificate Server Component

- Certificate Service
- Certificate Service Default Policy Module
- Certificate Service Default Exit Module

Embedded Component

- Bootable CD ROM Disk Image Driver
- Embedded System Setup
- Enhanced Write Filter Driver
- Enhanced Write Filter Formatter Library
- EFW Volume Creation and Configuration
- First Boot Agent
- First Boot Agent Registry Restoration Utility
- Reseal Run-Time Image
- System Stand-Alone Optional Component Manager
- Windows File Protection

Executive Component

- Executive Object Services
- 64 bit Kernel Debug Support
- Application Compatibility Support
- Cache Manager
- Configuration Manager
- Graphics Device Interface
- Hardware Abstraction Layer (HAL)
- Kernel Debug Manager
- Kernel Mode Windows Management Instrumentation
- Kernel Runtime
- Local Process Communication
- Memory Manager
- Microkernel
- Object Manager
- Plug and Play Manager
- Power Manager
- Process Manager
- Security Reference Monitor
- Virtual DOS Machine
- Window Manager (User)

Hardware Component

- IA 32 Hardware
- IA 64 Hardware

Windows Firewall Component

- Application Layer Gateway Service
- Home Networking Configuration Manager

- IP Network Address Translator
- IPv6 Firewall Driver
- MAC Bridge Driver
- Network Address Translation Helper

Internet Information Server Component

- Internet Information Services
- IIS CoAdmin
- IIS ISAPI Handler
- IIS Metadata DLL
- IIS Reset Control
- IIS Web Admin Service
- IIS Web Server Core
- IIS Worker Process
- ISAPI DLL for Web Printing
- Metadata and Admin Service
- URL Authorization ISAPI
- Web Application Manager Registration
- WebDAV ISAPI Extension and File Handle Cache
- WinHTTP Web Proxy Auto Discovery Service

IO:Core Component

- I/O Manager
- Kernel Security Device Driver
- File System Recognizer
- Mount Manager

IO: File Component

- CD-ROM File System
- Encrypting File System
- Fast FAT File System
- Mailslot Driver
- NPFS Driver
- NT File System Driver
- UDF File System Driver

IO: Net Component

- TCP/IP Protocol Driver
- Ancillary Function Driver for WinSock
- Browser
- Distributed File System Filter Driver
- General Packet Classifier Driver
- HTTP Driver
- IP Filter Driver
- IP in IP Encapsulation Driver
- IPSec Driver
- IPv6 Driver
- Loopback Network Driver
- NDIS 5.1 Wrapper Driver
- NDIS User Mode I/O Driver
- NetBT Transport Driver
- QoS Packet Scheduler Driver

Redirected Drive Buffering Subsystem Driver
Remote NDIS Miniport
Server Driver
SMB Mini Redirector
SMB Transport Driver
TDI Wrapper
WebDav Mini Redirector
Winsock2 IFS Layer Driver
Multiple UNC Provider and DFS Client Driver

IO: Devices Component

ACPI Driver
Adaptec adpu160m SCSI/RAID Controller
AGP 440 Bus Filter Driver
ATI ATI2MPAD Miniport Driver
ATI ATI2MTAG Miniport Driver
Audio Port Class Driver
BCM5703 Gigabit Ethernet
Beep Driver
Broadcom NetXtreme Gigabit Ethernet
Compaq Smart Array Controller SCSI Miniport Driver
Disk Manager I/O Driver
File System Filter Manager
FIPS Crypto Driver
Floppy Disk Controller Driver
Floppy Driver
FT Disk Driver
HID Class Library
HID Keyboard Filter Driver
HID Mouse Filter Driver
HID Parsing Library
i8042 Port Driver
IBM ServeRAID Adapter Storport Miniport Driver
IDE/ATAPI Port Driver
Intel e1000645 NIC Miniport Driver
Intel e100b645 NIC Miniport Driver
Intel Pro Adapter Driver
Intelligent I/O Miniport Driver
Intelligent I/O Utility Filter Driver
Keyboard Class Driver
LSI Logic PCI SCSI-FC MPI Miniport Driver
LSI Logic Symbios Ultra3 SCSI Miniport Driver
MegaRAID RAID Controller Driver
Mouse Class Driver
Null Driver
Parallel Port Driver
Partition Manager
Plug and Play PCI Enumerator
Plug and Play Software Device Enumerator Driver
PnP Disk Driver

- PnP ISA Bus Driver
- Processor Device Driver
- Redbook Audio Filter Driver
- SCSI CD-ROM Driver
- SCSI Class System Driver
- SCSI Floppy Driver
- SCSI Port Driver
- SCSI Tape Class Driver
- Serial Device Driver
- Serial Port Enumerator
- Smart Card Driver Library
- Storage Port Driver
- Update Driver
- USB 1.1 & 2.0 Port Driver
- USB CC Generic Parent Driver
- USB CCID Driver
- USB Host Controller Interface Miniport Drivers
- USB Mass Storage Driver
- USB Miniport Driver for Input Devices
- USB Root Hub Driver
- VGA/Super VGA Video Driver
- Video Port Driver
- Volume Shadow Copy Driver
- Watchdog Driver
- Windows Management Interface for ACPI

Net Support Component

- Domain Name Service
- COM+ Configuration Catalog Server
- COM+ Event System Service
- COM+ Services
- DHCP Service
- Distributed COM Services
- Internet Extensions for Win32
- IPv6 over IPv4 Service
- Network Connections Manager
- Network Location Awareness
- Routing Information Protocol for Internet Protocol
- RPC Endpoint Mapper
- RPC Locator
- Simple TCP/IP Services Service DLL
- TCP/IP NetBios Transport Services DLL
- TCP/IP Services Application
- Web DAV Service DLL

OS Support Component

- Distributed File System Service
- Logical Disk Manager Administrative Service
- Print Spooler
- Removable Storage Manager

Session Manager
WMI Provider Host
WMI Service

Security Component

Active Directory Replication Management
Core Directory Service
Credential Manager
Data Protection API
Directory Services Role Management
Encrypting File System Service
Inter-Site Messaging
IPSec SPD Server
KDC Service
Kerberos Security Package
LDAP
LSA Audit
LSA Authentication
LSA Policy
MAPI Based Directory Request
Microsoft Authentication Package v1.0
Microsoft Digest Access
Net Logon Service DLL
NT Directory Service Backup & Restore
Oakley Key Manager
PKI Trust Installation and Setup
Protected Storage Server
SAM Server
Secondary Logon Service
TLS / SSL Security Provider
TLS / SSL Service for HTTP
Trust Signing APIs

Services Component

Services and Controller App
Alert Service
Application Experience Lookup Service
Computer Browser Service DLL
Cryptographic Services
Event Logging Service
File Replication Service
Generic Host Process for Win32 Services
Logical Disk Manager Service
Non-COM WMI Event Provision APIs
NT Messenger Service
Remote Registry Service
Server Service DLL
SIS Groveler Service
Smart Card Resource Management Server
System Event Notification Service (SENS)

- User-mode Plug-and-Play Service
- Virtual Disk Service
- Volume Shadow Copy Service
- Windows Installer Service
- Windows Security Center Service
- Windows Security Configuration Editor Engine
- Windows Shell Services Dll
- Windows Time Service
- Workstation Service

WinLogon Component

- Windows Logon Application
- Auto Enrollment DLL
- Group Policy
- Group Policy Object Processing
- Infineon SICRYPT Smart Card CSP
- Profile Mapping
- Syskey
- Trust Verification APIs
- User Environment
- Windows File Protection

Win32 Component

- Client Server Runtime Process
- Base Server
- Windows Server DLL

Administrator Tools Component

- auditusr
- Auth_Mgr_GUI
- Backup_and_Restore
- Certification Authority GUI.
- Certreq Utility
- cipher.exe
- COM+ Applications
- Computer Management
- ConfigureYourServerWizard
- Date_and_Time
- DCOM Config
- Default_Group_Policy_Object_Restore_Utility
- Device Manager
- DHCP Snap-in
- Disk Management
- Disk Quota GUI
- DNS Snap-in
- Domains and Trusts
- Driver_Verifier_Manager
- efsadu
- Event Viewer
- Explorer
- Group Policy Spec

Group Policy Refresh
Group Policy
GUI Component
IIS_Manager_GUI
IPSec Settings GUI Spec
IPv6 Monitor DLL
Keyring Manager
Network
NetworkID
OU Delegation
Printers GUI
Registry_Editor
RSoP
SAM Lock Tool
SCWcmd.exe
Security Config Editor
Security Policy GUI
Security_Config_Wizard
Services
Session_Locking
Share_a_Folder_Wizard
Sigverif Design Spec
Sites and Services
Users and Groups
Volume Shadow Copy Service Command Line
Windows Mngt Infra (WMI)
Windows_Firewall_GUI