Windows 2000 Security Target

ST Version 2.0

18 October 2002

Prepared For:

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

Prepared By:

Science Applications International Corporation
7125 Gateway Drive
Columbia, MD 21046

ST Master, 10/18/02, Rev 1.9

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 CC Identification

ST Title – Microsoft Windows 2000 Security Target

ST Version – Version 2.0

TOE Software Identification – The following Windows 2000 Operating Systems:

- Windows 2000 Professional with Service Pack 3 (SP3) with Q326886 Hotfix
- Windows 2000 Server with SP3 with Q326886 Hotfix, or
- Windows 2000 Advanced Server with SP3 with Q326886 Hotfix

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

- Compaq Proliant ML570
- Compaq Proliant ML330 (both 2-processor and 4-processor version)
- Compaq Professional Workstation AP550
- Dell Optiplex GX400
- Dell PE 2500
- Dell PE 6450
- Dell PE 2550
- Dell PE 1550

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

Common Criteria Identification – Common Criteria for Information Technology Security Evaluation, Version 2.1, August 1999.

International Standard – ISO/IEC 15408:1999.

Keywords – operating system, sensitive data protection device, directory service, network management, desktop management, single sign on, discretionary access control, security target, CAPP, EAL 4, Microsoft Windows.

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

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional requirements, Version 2.1, August 1999, extended (Part 2 extended)
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.1, August 1999, conformant, Evaluation Assurance Level 4 (Part 3 augmented).

1.3 Strength of Environment

The evaluation of Windows 2000 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 were chosen to be consistent with this goal and to be compliant with the Controlled Access Protection Profile (CAPP). The TOE assurance level is Evaluation Assurance Level (EAL) 4 augmented with ALC_FLR.3 and the TOE minimum strength of function 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:

- Security Functional Requirements – 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 interaction 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.
- Discretionary Access Control (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 Security Target Overview and Organization

The Windows 2000 Target of Evaluation (TOE) is a general-purpose, distributed, network operating system that provides controlled access between subjects and user data objects. Windows 2000 has a broad set of security capabilities including single network logon; access control and data encryption; extensive security audit collection; and Light-weight Directory Access Protocol (LDAP) Directory-based resource management. The Windows 2000 TOE provides the following security services: user data protection, audit, identification and authentication, security management, protection of the TOE Security Functions (TSF), resource quotas and TOE access banners. The Windows 2000 security policies provide network-wide controlled access protection (access control), encrypted data/key protection, and encrypted file protection. These policies enforce access limitations between individual users and data objects. The TOE is capable of auditing security relevant events that occur within a Windows 2000 network. All these security controls require users to identify themselves and be authenticated prior to using any node on the network.

The Windows 2000 ST contains the following additional sections:

- TOE Description (Section 2) – Provides an overview of the TOE security functions 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.
- Protection Profile 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 as to their consistency, completeness and suitability.

2. TOE Description

The TOE includes the Windows 2000 operating system, supporting hardware, and those applications necessary to manage, support and configure the operating system.

2.1 Product Types

Windows 2000 is a preemptive multitasking, multiprocessor, and multi-user operating system. In general, operating systems provide users with a convenient interface to manage underlying hardware. They control the allocation and manage computing resources such as processors, memory, and Input/Output (I/O) devices. Multi-user operating systems, such as Windows 2000, 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 2000 provides an interactive user interface, as well as a network interface. The TOE includes a homogenous set of Windows 2000 systems that can be connected via their network interfaces and may be organized into domains. A domain is a logical collection of Windows 2000 systems that allows the administration and application of a common security policy and the use of a common accounts database. Windows 2000 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 2000 system, whether it is a DC server, non-DC server, or workstation, is part of the TOE and provides a subset of the TOE Security Functions (TSFs). The TSF for Windows 2000 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 operating system, Windows 2000 can also be categorized as the following types **Information Assurance (IA)** or IA enabled Information Technology (IT) products:

Windows 2000 is a **Sensitive Data Protection Device** to defend the Computing Environment. The core mechanism in this case is the Windows 2000 Encrypting File System, which is part of the Windows 2000 TOE.

Windows 2000 is a **Directory Service** product to support Security Infrastructure. The LDAP based access and management of Windows Active Directory objects is part of the Windows 2000 TOE Security Function Interfaces (TSFI).

Windows 2000 is a **Network Management** product to support the Security Infrastructure. Windows 2000 Group Policy is part of the Windows 2000 TOE and provides the network management in Windows 2000 networks.

Windows 2000 is a **Desktop Management** product to support the Security Infrastructure. Windows 2000 Group Policy Service, which is part of Windows 2000 TOE, provides the desktop management of Windows 2000 TOE desktops.

Windows 2000 is a **Single Sign On** product for Windows 2000 networks to defend the Computing Environment. Windows 2000 supports single sign on to the TOE.

Additionally, the Windows 2000 Internet Protocol Security (IPSec) Service and its associated Transport Driver Interface (TDI) based network support, which is part of the Windows 2000 TOE, can be used to support **Virtual Private Network (VPN)** functionality.

2.2 Product Description

Windows 2000 is an operating system that supports both workstation and server installations. The TOE includes three product variants of Windows 2000: Professional, Server, and Advanced Server. The server products additionally provide Domain controller features including the Active Directory and Kerberos Key Distribution Center. Otherwise, all three 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 2000 Professional is suited for business desktops and notebook computers; it is the workstation product. Windows 2000 Server is designed for workgroups and small business environments. Windows 2000 Advanced Server includes availability and scalability features that support higher volumes of users and more complex applications.

The security features addressed by this security target are those provided by Windows 2000 as an operating system. Microsoft provides several Window 2000 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 services; certificate authority services; web based applications; and firewall functionality.

2.3 Product Features

Windows 2000 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 2000 Administration and Management Features

Windows 2000 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 2000 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 Active Directory 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 2000 offers groups that are pre-configured with specific user rights and/or privileges. These groups are referred to as “built-in groups.” The Windows 2000 built-in groups fall into 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 2000 introduces Active Directory, a scalable, standard-compliant directory service. Active Directory centrally manages Windows-based clients and servers, through a single consistent management interface, reducing redundancy and maintenance costs.

Active Directory 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.

Active Directory can interoperate or synchronize data with other directory services using Lightweight Directory Access Protocol (LDAP), Microsoft Directory Service Synchronization, or Active Directory Connector.

Access Control Lists
Windows 2000 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), the actions an individual can take on an object, and the events that are audited.

Every object has a unique security descriptor that includes an Access Control List (ACL). An ACL is a list of entries that grant or deny specific access rights to individuals or groups. The Windows 2000 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 2000 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
Windows Management Instrumentation (WMI) is a uniform model through which management data from any source can be managed in a standard way. WMI provides this for software, such as applications, while WMI extensions for the Windows Driver Model (WDM) provide this for hardware or hardware device drivers.

Administrative Tools
Windows 2000 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 2000 backup and recovery features make it easier to backup data and then recover data in the event of a hard disk failure. Windows 2000 allows back up to a single file on a hard disk and tape media.

Administrative wizards: Windows 2000 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
Microsoft Management Console 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 2000 are available through the Microsoft Management Console (MMC) snap-ins (including Active Directory Domains and Trusts, Active Directory Sites and Services, Active Directory Users and Computers, Component Services, Computer Management, Certificate Management, Event Viewer, Group Policy, IP security 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 2000 identifies device drivers that have passed the Windows Hardware Quality Labs test and warns users if they are about to install an uncertified driver.

2.3.2 Windows 2000 Network Security Features

Windows 2000 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 Internet Protocol Security (IPSec). Several features that support improved network security are briefly described below.

Encrypting File System (EFS)
Windows 2000 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 protocol Windows 2000 provides fast, single sign-on to Windows 2000-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 2000 builds secure network sites using the latest standards, including 56-bit and 128-bit SSL/TLS, IPSec, Server Gated Cryptography; and Kerberos v5 authentication.

Secure network communications
Windows 2000 supports end-to-end encrypted communications across network using the IPSec standard. It protects sensitive internal communications from intentional or accidental viewing. Active Directory provides central policy control for its use to make it deployable.

Crypto API
Windows 2000 Crypto API provides applications access to FIPS-140-1 compliant cryptographic functions, public keys, credential management and certificate validation functions.

Dynamic DNS
The Active Directory integrated, Internet standards-based Domain Name System (DNS) service simplifies object naming and location through Internet protocols, and improves scalability, performance and interoperability. Systems that receive addresses from a Dynamic Host Configuration Protocol (DHCP) server are automatically registered in DNS. Replication options through Active Directory can simplify and strengthen name replication infrastructure.

2.3.3 Windows 2000 Scalability Features

Windows 2000 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 2000 Professional and Windows 2000 Server support up to 4 gigabytes (GB) of RAM and up to two symmetric multiprocessors. Windows 2000 Advanced Server takes advantage of larger amounts of memory to improve performance and handle the most demanding applications, with support for up to 8 gigabytes (GB) of RAM with Intel's Physical Address Extension (PAE).

8-way Symmetric Multi-Processor Support
Windows 2000 Advanced Server can be scaled up by using the latest 8-way SMP servers for more processing power. Windows 2000 Server delivers support for up to 4-way SMP servers.

High throughput and bandwidth utilization
With support for up to 1 GB networks, Windows 2000 Server delivers high performance processing on high performance networks. Increased throughput increases performance without having to increase network bandwidth.

Job Object API
The Windows 2000 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.

Distributed File System (DFS)
Windows 2000 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.

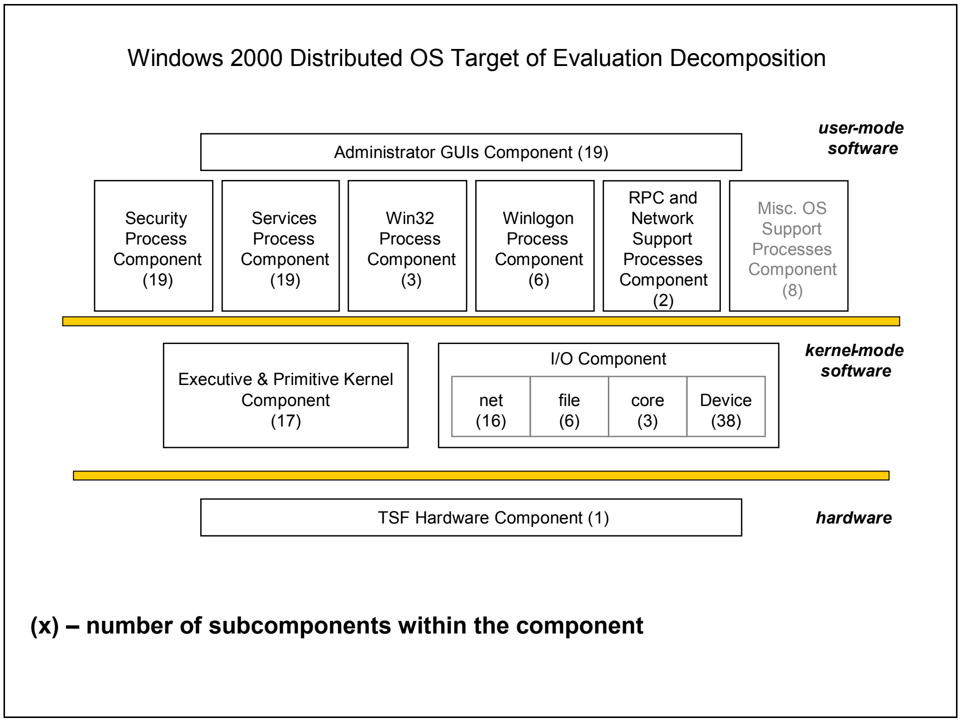
Multi-master Replication
Active Directory uses multi-master replication to ensure high scalability and availability in distributed network configurations. "Multi-master" means that each directory replica in the network is a peer of all other replicas; changes can be made to any replica and will be reflected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

2.4 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 2000 that comprise the TOE. The components/subcomponents are large portions of the Windows 2000 operating system, and generally fall along process boundaries and a few major subdivisions of the kernel mode operating system.


Windows 2000 Distributed OS Target of Evaluation Decomposition
user-mode software
Administrator GUIs Component (19)
Security Process Component (19)
Services Process Component (19)
Win32 Process Component (3)
Winlogon Process Component (6)
RPC and Network Support Processes Component (2)
Misc. OS Support Processes Component (8)
kernel-mode software
Executive & Primitive Kernel Component (17)
I/O Component
net (16)
file (6)
core (3)
Device (38)
TSF Hardware Component (1)
hardware
(x) – number of subcomponents within the component

The system components are:

- Security: This component is the user-mode process that includes all security management services and functions. In the case where the process is running on a domain controller, it also includes the Kerberos key distribution center services and Active Directory services.
- I/O Component: This is the kernel-mode software that implements all I/O related services (including file systems and networking), as well as all driver-related services.
- Executive & Primitive Kernel: This component consists of
  - the kernel-mode software that provides the core operating system services to include memory management, process management, and inter-process communication; and
  - the primitive kernel of the low-level kernel-mode software that only provides services to other kernel-mode software.

  This component implements the non-I/O TOE security function interfaces for the kernel-mode.
- Winlogon: This process provides interactive logon services, as well as screen locking and trusted path.
- Win32: This process provides various support services for Win32 applications and the command console application running in the default Windows environment.
- Services: This component is the service controller process and the services controlled via its interfaces.
- OS Support: This component is a set of processes that provide various other operating system support functions and services such as the print spooler and remote Registry interfaces.
- Network Support: These processes contain various support services for Remote Procedure Call (RPC), COM/DCOM, and other network services.
- Administrator GUIs: Applications used by an authorized administrator to securely administer the system.
- TOE Hardware: This component includes all hardware used by the system to include the processor(s), motherboard and associated chip sets, controllers, and I/O devices.

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 machine or equivalent processor (including Pentium family) with up to 4 CPUs for a Server product and up to 8 CPUs for the Advanced Server 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, and
- Network Adaptor.

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 2000 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 2000 requires each user to be identified and authenticated prior to performing any functions. An interactive user invokes a trusted path in order to protect his identification and authentication information. Windows 2000 maintains a database of accounts including their identities, authentication information, group associations, and privilege and logon rights associations. Windows 2000 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 2000 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 2000 enforces discretionary access control policy and functions; crypt access control policy and functions; encrypting file system policy; and, object and subject residual information protection. Windows 2000 uses access control methods to allow or deny access to objects, such as files, directory entries, and printers. It authorizes access to these resource objects through the use of security descriptors, which are sets of information identifying users and their specific access to resource objects. Windows 2000 provides additional access control protection for user data through the use of data encryption mechanisms. These mechanisms only allow authorized users access to encrypted data. Windows 2000 also protects user data by ensuring that resources exported to user-mode processes do not have any residual information.
- **Protection of TOE Security Functions** – Windows 2000 provides a number of features to ensure the protection of TOE security functions. Windows 2000 protects against unauthorized data disclosure and modification by using a suite of Internet standard protocols including Internet Protocol Security (IPSEC) and Internet Security Association and Key Management Protocol (ISAKMP). Windows 2000 ensures process isolation security for all processes through private virtual address spaces, execution context and security context. The Windows 2000 data structures defining process address space, execution context, and security context are stored in protected kernel-mode memory.

- **Resource Utilization –** Windows 2000 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 2000 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 2000 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 2000. The assumptions and policies are primarily derived from the Controlled Access Protection Profile (CAPP), while the threats have been introduced to better represent specific threats addressed by Windows 2000.

3.1 Threats to Security

Table 3-1 presents known or presumed threats to protected resources that are addressed by Windows 2000.

Table 3-1 Threats Addressed by Windows 2000

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.

3.2 Organizational Security Policies

Table 3-2 describes organizational security policies that are addressed by Windows 2000.

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**

Security Policy	Description	PP Source
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 2000 is intended to be used. This includes assumptions about the connectivity, personnel, and physical aspects of the environment.

Windows 2000 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 2000 is a distributed system connected via network media. It is assumed that the following connectivity conditions 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 following personnel conditions 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 management 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 2000 is intended for application in user areas that have physical control and monitoring. It is assumed that the following physical conditions 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 2000 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 IT Security Objectives

The following are the Windows 2000 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 system data in transmission 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.	

4.2 Non-IT Security Objectives

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. These Non-IT Security Objectives are listed below.

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 security functional requirements (SFRs) for the TOE. This section organizes the SFRs by CC class.

Requirement Operations:

Within the text of each SFR taken from the CAPP, operations completed in the CAPP operations are underlined.

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

Within the text of each SFR taken directly from the CC and not included in the CAPP, the selection and assignment operations are bracketed in this ST, and refinement operations are bolded in this ST.

Within the text of each SFR, iteration operations that are not included in the CAPP are italicized in this ST. Additionally, iterated requirements are indicated by a letter in parenthesis placed at the end of the component short name and element name(s).

Strength of TOE Security Functions (SOF):

This ST includes the Strength of TOE Security Functions 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
- 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
- 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 functional requirement 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[1]
FAU_GEN.2	User Identity Association	None	None
FAU_SAR.1	Audit Review	Assignment	None
FAU_SAR.2	Restricted Audit Review	None	None
FAU_SAR.3	Selectable Audit Review	Assignment, Selection	Assignment, Selection
FAU_STG.1	Protected Audit Trail Storage[2]	Selection	None
FAU_STG.3	Action in Case of Possible Audit Data Loss	Assignment	Assignment
FAU_STG.4	Prevention of Audit Data Loss	Selection	Assignment
FDP_ACC.1	Discretionary Access Control Policy	Assignment, Refinement	Assignment
FDP_ACF.1	Discretionary Access Control Functions	Assignment, Refinement	Assignment, Refinement
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 $_{\text{(SOF)}}$[4]	Verification of Secrets[5]	Assignment	Refinement
FIA_UAU.7	Protected Authentication Feedback	Assignment	None
FIA_USB.1_EX[6]	User-Subject Binding	Extension	Assignment
FMT_MSA.1 (a)	Management of Object Security Attributes	Assignment, Selection	Assignment

[1] This requirement is refined beyond the CAPP to include additional auditable events.

[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.

[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".

CAPP Component		CAPP Operation	Additional ST Operations
		Selection	
FMT_MSA.3	Static Attribute Initialization	Assignment, Selection	Assignment
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	None
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
FMT_SMR.1	Security Roles	Assignment	Assignment
FPT_RVM.1	Non-bypassability of the TSP[7]	None	None
FPT_SEP.1	TSF Domain Separation[8]	None	None
FPT_STM.1	Reliable Time Stamps	None	None

Table 5-2 CC Components and Operations

CC Component	Component Name	ST Operations
FCS_COP.1	Cryptographic Operation	Assignment
FIA_AFL.1	Authentication Failure Handling	Assignment
FIA_UAU.2 $_{\text{(SOF)}}$[9]	User Authentication Before Any Action	None
FIA_UID.2	User Identification Before Any Action	None
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,

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

[8] This title is consistent with the CC. The CAPP title for this requirement is “Domain Separation” which is inconsistent with the CC.

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

		Selection, Iteration
FMT_MSA.1(b)	Management of Security Attributes	Assignment, Selection, Iteration
FMT_MTD.1(e)	Management of Account Lockout Duration	Assignment, Selection, Iteration
FMT_MTD.1(f)	Management of Minimum Password Length	Assignment, Selection, Iteration
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.2	Management of Unsuccessful Authentication Attempts Threshold	Assignment
FMT_SAE.1	Timed–limited Authorization	Assignment
FMT_SMR.3	Assuming Roles	Assignment
TRANSFER_PROT_EX	Internal TSF Data Transfer Protection	Extension
REPLICATION_EX	Internal Replication	Extension
FRU_RSA.1	Maximum Quotas	Assignment, Selection
FTA_SSL1	TSF-initiated Session Locking	Assignment
FTA_SSL.2	User-initiated Session Locking	Assignment
BANNERS_EX	Configurable TOE Access Banners	Extension
FTA_TSE.1	TOE Session Establishment	Assignment
FTP_TRP.1	Trusted Path	Assignment, Selection

5.1.1 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	None	
FAU_STG.1	None	
FAU_STG.3	Actions taken due to exceeding of a threshold	
FAU_STG.4	Actions taken due to the audit storage failure	
FDP_ACC.1	None	
FDP_ACF.1	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[10]	The use of the authentication mechanism	
FIA_UAU.7	None	
FIA_UID.1[11]	All use of the user identification mechanism, including the identity provided during successful attempts	The origin of the attempt (e.g. terminal identification).

[10] This requirement is not included in this Security Target, however, FIA_UAU.2 is which is hierarchical to FIA_UAU.1. The content for this audit event is contained in the audit event captured for FIA_UAU.2 described in Table 5-4.

Component	Event	Details
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	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	

Table 5-4 Other Auditable Events

Component	Event
FIA_AFL.1	Account locked out due to exceeding the maximum number of unsuccessful logon attempts
FIA_UAU.2	The use of the authentication mechanism
FIA_UID.2	All use of the user identification mechanism, including the identity provided during successful 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	IPSEC policy changes
FTA_SSL1	Attempt to unlock
FTA_SSL.2	Attempt to unlock
FTA_TSE.1	Logon Failure due to password expiration

[11] This requirement is not included in this Security Target, however, FIA_UID.2 is which is hierarchical to FIA_UID.1. The content for this audit event is contained in the audit event captured for FIA_UID.2 described in Table 5-4.

FTP_TRP.1	Authentication and unlocking attempts

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 user to interpret the information.

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 (FAU_SAR.3)

5.1.1.5.1 FAU_SAR.3.1

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

a) User identity;

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

5.1.1.6 Protected Audit Trail Storage (FAU_STG.1)

5.1.1.6.1 FAU_STG.1.1

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

5.1.1.6.2 FAU_STG.1.2

The TSF shall be able to prevent modifications to the audit records.

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

5.1.1.7.1 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].

5.1.1.8 Prevention of Audit Data Loss (FAU_STG.4)

5.1.1.8.1 FAU_STG.4.1

The TSF shall be able to prevent auditable events, except those taken by the authorized administrator, and [no other actions] if the audit trail is full.

5.1.2 Cryptographic Support (FCS)

5.1.2.1 Cryptographic Operation (FCS_COP.1)

5.1.2.1.1 FCS_COP.1.1

The TSF shall perform [the CryptoProtect encryption function and the CryptUnprotect decryption function] in accordance with a specified cryptographic algorithm [DES or Triple DES] and key size [56-bits or 168-bits] that meet the following [FIPS 140-1 Level 1].

5.1.3 User Data Protection (FDP) Requirements

5.1.3.1 Discretionary Access Control Policy (FDP_ACC.1)

5.1.3.1.1 FDP_ACC.1.1

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

[Named objects –

Desktop, 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, and Active Directory objects]; and all operations among subjects and objects covered by the DAC policy.

5.1.3.2 Discretionary Access Control Functions (FDP_ACF.1)

5.1.3.2.1 FDP_ACF.1.1

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) 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
    - For directory service (DS) object entries, globally unique identifiers (GUID) indicating DS-specific object attributes.
- 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.

].

5.1.3.2.2 FDP_ACF.1.2

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 or modify the object's DACL, or the subject is the owner and the operation is to create an object

]

5.1.3.2.3 FDP_ACF.1.3

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.

].

5.1.3.2.4 FDP_ACF.1.4

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.

].

5.1.3.3 Object Residual Information Protection (FDP_RIP.2)

5.1.3.3.1 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.

5.1.3.4 Subject Residual Information Protection (Note1_EX)

5.1.3.4.1 FDP_RIP.2.Note1.1

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 authorized administrator specified number of] unsuccessful authentication attempts occur related to [any 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 Keys, Privileges, and Logon Rights]

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 **250,000,000,000,000;**
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 **25,000,000,000,000**; and
c) Any feedback given during an attempt to use the authentication mechanism will not reduce the probability below the above metrics.

5.1.4.4 User Authentication Before Any Action (FIA_UAU.2)

5.1.4.4.1 FIA_UAU.2.1

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

5.1.4.5 Protected Authentication Feedback (FIA_UAU.7)

5.1.4.5.1 FIA_UAU.7.1

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

5.1.4.6 User Identification Before Any Action (FIA_UID.2)

5.1.4.6.1 FIA_UID.2.1

The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on the behalf of that user.

5.1.4.7 User Subject Binding (FIA_USB.1_EX)

5.1.4.7.1 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 identity which is associated with auditable events;
b) The user identity or identities which are used to enforce the Discretionary Access Control Policy;

c) The group membership or memberships used to enforce the Discretionary Access Control Policy;
d) [Privileges.]

5.1.4.7.2 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.]

5.1.4.7.3 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 TSF 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 Object Security Attributes (FMT_MSA.1(a))

5.1.5.4.1 FMT_MSA.1.1(a)

The TSF shall enforce the Discretionary Access Control Policy to restrict the ability to modify the 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].

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

5.1.5.5.1 FMT_MSA.1.1(b)

The TSF shall enforce the [Discretionary Access Control Policy] to restrict the ability to [delete] the security attributes [encryption policy attributes associated with a file] 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].

5.1.5.6 Static Attribute Initialization (FMT_MSA.3)

5.1.5.6.1 FMT_MSA.3.1

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.

5.1.5.6.2 FMT_MSA.3.2

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

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

5.1.5.7.1 FMT_MTD.1.1(a)

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

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

5.1.5.8.1 FMT_MTD.1.1(b)

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

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

5.1.5.9.1 FMT_MTD.1.1(c)

The TSF shall restrict the ability to initialize and modify the user security attributes, other than authentication data to authorized administrators.

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

5.1.5.10.1 FMT_MTD.1.1(d)

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

5.1.5.10.2 FMT_MTD.1.1(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.

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

5.1.5.11.1 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].

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

5.1.5.12.1 FMT_MTD.1.1(f)

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

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

5.1.5.13.1 FMT_MTD.1.1(g)

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

5.1.5.14 Management of NTFS Volume Quota Settings (FMT_MTD.1(h))

5.1.5.14.1 FMT_MTD.1.1(h)

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

5.1.5.15 Management of Advisory Warning Message (FMT_MTD.1(i))

5.1.5.15.1 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].

5.1.5.16 Management Audit Log Size (FMT_MTD.1(j))

5.1.5.16.1 FMT_MTD.1.1(j)

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

5.1.5.17 Management of User Inactivity Threshold (FMT_MTD.1(k))

5.1.5.17.1 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].

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

5.1.5.18.1 FMT_MTD.2.1

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

5.1.5.18.2 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].

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

5.1.5.19.1 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.

5.1.5.19.2 FMT_REV.1.2(a)

The TSF shall enforce the rules:

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

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

5.1.5.20.1 FMT_REV.1.1(b)

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

5.1.5.20.2 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].

5.1.5.21 Time-limited Authorization (FMT_SAE.1)

5.1.5.21.1 FMT_SAE.1.1

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

5.1.5.21.2 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 attribute has passed.

5.1.5.22 Security Roles (FMT_SMR.1)

5.1.5.22.1 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
d) [No additional roles].

5.1.5.22.2 FMT_SMR.1.2

The TSF shall be able to associate users with roles.

5.1.5.23 Assuming Roles (FMT_SMR.3)

5.1.5.23.1 FMT_SMR.3.1

The TSF shall require an explicit request to assume the following roles [authorized administrator].

5.1.6 Protection of the TOE Security Functions (FPT)

5.1.6.1 Internal TSF Data Transfer Protection (TRANSFER_PROT_EX)

5.1.6.1.1 TRANSFER_PROT_EX.1

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

5.1.6.2 Internal Replication (REPLICATION_EX)

5.1.6.2.1 REPLICATION_EX.1

The TOE shall ensure that changes to TSF data are copied between parts of the TOE, and the receiving parts of the TOE shall accept the TSF data only when the TSF data is more recent than the current values its TSF data.

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

5.1.6.3.1 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.

5.1.6.4 TSF Domain Separation (FPT_SEP.1)

5.1.6.4.1 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.

5.1.6.4.2 FPT_SEP.1.2

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

5.1.6.5 Reliable Time Stamp (FPT_STM.1)

5.1.6.5.1 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 TSF-Initiated Session Locking (FTA_SSL.1)

5.1.8.1.1 FTA_SSL.1.1

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

5.1.8.1.2 FTA_SSL.1.2

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

5.1.8.2 User-Initiated Session Locking (FTA_SSL.2)

5.1.8.2.1 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.

5.1.8.2.2 FTA_SSL.2.2

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

5.1.8.3 Configurable TOE Access Banners (BANNERS_EX)

5.1.8.3.1 BANNERS_EX.1

Before establishing a user session, the TSF shall be able to display an advisory warning message regarding unauthorized use of the TOE.

5.1.8.4 TOE Session Establishment (FTA_TSE.1)

5.1.8.4.1 FTA_TSE.1.1

The TSF shall be able to deny session establishment based on [authentication data expiration].

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 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 and session unlocking].

5.2 TOE Security Assurance Requirements

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

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.4.4C

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

5.2.1.2.8 ACM_CAP.4.5C

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

5.2.1.2.9 ACM_CAP.4.6C

The CM system shall uniquely identify all configuration items.

5.2.1.2.10 ACM_CAP.4.7C

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

5.2.1.2.11 ACM_CAP.4.8C

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

5.2.1.2.12 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.

5.2.1.2.13 ACM_CAP.4.10C

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

5.2.1.2.14 ACM_CAP.4.11C

The CM system shall support the generation of the TOE.

5.2.1.2.15 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.

5.2.1.2.16 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 CM documentation.

5.2.1.3.2 ACM_SCP.2.1C

The CM documentation shall show that the CM system, as a minimum, tracks the following: the TOE implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws.

5.2.1.3.3 ACM_SCP.2.2C

The CM documentation shall describe how configuration items are tracked by the CM system.

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 documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.

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 documents 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.1.1E[12]

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

[12] This label is consistent with the CAPP. In the CC, this element is incorrectly labeled as "ATE_DPT.1.2E"

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.

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 and document an analysis of the TOE deliverables searching for ways in which a user can violate the TSP.

5.2.7.3.2 AVA_VLA.2.2D

The developer shall document the disposition of identified vulnerabilities.

5.2.7.3.3 AVA_VLA.2.1C

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

5.2.7.3.4 AVA_VLA.2.2C

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

5.2.7.3.5 AVA_VLA.2.1E

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

5.2.7.3.6 AVA_VLA.2.2E

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

5.2.7.3.7 AVA_VLA.2.3E

The evaluator shall perform an independent vulnerability analysis.

5.2.7.3.8 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.

5.2.7.3.9 AVA_VLA.2.5E

The evaluator shall determine that the TOE is resistant to penetration attacks performed by an attacker possessing a low 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

This chapter describes the Windows 2000 security functions and associated assurance measures. The Windows 2000 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 2000 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 TOE security functions (TSFs) and a mapping of security functions to security functional requirements. The TOE performs the following security functions:

- Audit;
- User Data 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,
- 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: The audit log includes a field none as “type.” This type is different from the general term type used in Section 5.1 for FAU requirements. In the audit log, an event record can be one of five types: Information, Warning, Error, Success Audit, or Failure Audit. The latter two are the only types used in the security log, indicating 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)

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. The exception is that the Event Logger itself records one event record, when the security log is cleared.

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.

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;FMT_MSA.1(a); FMT_MSA.3; FMT_REV(b)
Privilege Use	Audits attempts to use security relevant privileges. Security relevant privileges are those privileges that are related to the TOE Security Functions 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
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; 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
Directory Service Access	Audit access to directory service objects and associated properties.	FDP_ACF.1
Logon	Audit attempts to logon or logoff the system, attempts to make a network connection.	FIA_SOS.1; FIA_UAU.2; FIA_UID.2; 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 domain controller receives a logon request.	FIA_SOS.1; FIA_UAU.2; FIA_UID.2;

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.

6.1.1.3 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.

6.1.1.4 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 log is a system resource, created during system startup. No interfaces exist to create, destroy, or change security of event logs. The LSA subsystem is the only service registered to enter events into the security log.

SFR Mapping

The **Audit function** satisfies the following security functional requirements:

- 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 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 – 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 System access control lists (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.
- FAU_STG.1 – The interface to the security log is limited by the event logger. Only authorized administrators can view the audit data.
- 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.

6.1.2 User Data Protection Function

The user data protection security services provided by the TOE are:

- Discretionary Access Control;
- Cryptographic Protection
- Residual Data Protection.

6.1.2.1 Discretionary Access Control (DAC)

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

Table 6-2 Named Objects

Name	Description
Desktop	The primary object used for graphical displays. The interactive window station has three default desktops created by WinLogin.
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 filesystem 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.

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 information in the token includes: the security identifier (SID) for the user, SIDs representing groups for which the user is a member, privileges assigned to the user, an owner SID 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 Identification and Authentication 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 exits).

6.1.2.1.2 Object DAC Attributes

Security Descriptors (SDs) contain all of the security attributes associated with an object. All objects in Table 6-2 have an associated security descriptor. The security attributes from a SD used for access control are the object owner SID, the discretionary access control list (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 Security Reference Monitor (SRM) makes all access control decisions.

For some objects (in particular, DS objects), the TSF does not maintain an opened context (i.e., 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 globally unique ID (GUID). The TOE allows access control for DS objects to the level of GUIDs (i.e., the entire DS object, a given property set, and or a specific property). Like all objects, DS objects still have a single security descriptor for the entire object; however the DACL for a DS object can contain ACEs the grants/denies access to any of the associated GUIDs.

6.1.2.1.3 DAC Enforcement Algorithm

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

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

1. Privilege Check –
   a. Check for SeSecurity privilege – This is required if ACCESS_SYSTEM_SECURITY is in the desired access mask. If ACCESS_SYSTEM_SECURITY is requested and the requestor does not have this privilege, access is denied. Otherwise ACCESS_SYSTEM_SECURITY is granted.

   b. Check for SeTakeOwner privilege – If the desired mask has WRITE_OWNER access right, and the privilege is found in the requestor's token, then WRITE_OWNER access is granted.

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 encryption key (FEK). The FEK is used to encrypt the file data using the Triple DES 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 by one of the following methods:

- If a security descriptor is explicitly provided as part of the create call, the DACL is initialized from the provided SD;
- If a SD is not provided, the DACL is initialized based on ACEs from a parent object that have inheritance attributes that indicate the ACE is applicable to children objects. In case of DS objects, the object class default SD is merged with the inherited ACEs ; or,
- If there is not a parent object from which to inherit ACEs, the DACL is initialized using the default DACL in the subject's token.

All tokens are created with an appropriate default DACL, which can be applied to the new objects as appropriate. The default DACL is for the SYSTEM SID and the user SID to have all access. The SYSTEM SID is a special SID representing TSF trusted processes.

6.1.2.2 Cryptographic Protection

The TOE provides the ability to encrypt data blocks and protect the data from being decrypted using the CryptoProtect function. The TSF uses this function to encrypt the private keys used by the RSA public-key-based encryption algorithm to decrypt the FEKs (FEK are described the DAC Enforcement of Encrypted Files). However, this function is also available to users to encrypt 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 user's master key is protected by the TSF based on SHA-1 Hash-Based Message Authentication Code (HMAC). The protection 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 master key backup key is also calculated based on a backup/restore key of a Domain Controller (DC), if present. The key size is based on the DES or Triple DES 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. Only an authorized administrator can select the algorithm that is used for encryption and decryption and only the administrator can select the key size. The algorithm chosen and key size chosen is specified by a registry key that is restricted to only authorized administrators.

Upon a request to decrypt data provided by the user, the TSF performs the CryptUnprotect function using the user's password to get the user's master key. It then uses the user's master key to decrypt the data. At that point, the TSF performs a MAC (Message Authentication Code) check to verify proper decryption. If the MAC check succeeds, then the data is returned. If the MAC check fails, then the TSF attempts to decrypt the data using the restore key. This may be necessary if the password has changed. The TSF decrypts the data using the restore key and then verifies that the user is authorized to the data by comparing the SID extracted using the backup/restore master key to the SID of the caller. If the caller is authorized to the data (i.e., the SIDs match), then the decrypted data is returned.

The encryption and decryption operations are performed by independent modules, known as cryptographic service providers (CSPs). The CSPs are FIPS 140-1 Level 1 compliant. The FIPS 140-1 standard was

developed by the National Institute of Standards and Technology (NIST). FIPS 140-1, titled "Security Requirements for Cryptographic Modules," specifies the United States government's requirements for proper design and implementation of hardware and software cryptographic modules that perform cryptographic operations for sensitive but unclassified information. FIPS 140-1 has been adopted by the Canadian Communication Security Establishment and the American National Standards Institute.
FIPS 140-1 is widely regarded as a de facto standard for cryptographic modules.

6.1.2.3 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.[13] 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 re-initialized when a thread context switch occurs.

SFR Mapping

The **User Data Protection function** satisfies the following security functional requirements:

- FDP_ACC.1 - The Security Reference Monitor (SRM) mediates all access to objects, including kernel-based objects and user-mode TSF server-based objects. All access to objects is predicated on the SRM validating the access request. In the case of most objects, this DAC validation is performed on initial access (e.g., “open”) and subsequent use of the object is via a handle that includes a granted access mask. For some objects (in particular DS objects), every reference to the object requires a complete DAC validation to be performed. The TSF mediates read access by subjects to encrypted files by protecting user and recovery private keys and using those keys to protect the FEK.

- FDP_ACF.1 – 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 file encryption keys associated with the file and protected using authorized users’ private keys to protect the encrypted file contents.

- 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.

[13] For APIs that create objects, the caller may provide data to initialize the object.

  - 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.3 - The TSF provides restrictive default values for security attributes used to provide access control via the process's default DACLs. Users who create objects can specify a SD with a DACL to override the default. The initial keys are cryptographically generated and cannot be modified.

- FMT_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.

- FCS_COP.1 – The TSF uses the DES or Triple DES (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.

- FMT_MSA.1(b) – The TSF associates private keys with users. Only the owner of the private key or an administrator can delete these keys.

- 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 re-initialized when threads are created or when a context switch occurs.

6.1.3 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, regardless of whether the user is logging on interactively or is accessing the system via a network connection. The only 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.

6.1.3.1 Logon Type

The TOE supports four types of user logon: interactive ("Logon locally"), network ("Access this computer from Network"), batch ("Logon as a batch job"), and service ("Logon as a service"). 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 – Common Object Model - servers). The service logon type is used when a service process is started to provide a user context in which that service will operate.

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.

6.1.3.2 Trusted Path

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).

6.1.3.2.1 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.

6.1.3.3 User Attribute Database

6.1.3.3.1 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 (Security Identifier) – a user identifier 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);
- Private Keys – used to decrypt user's File Encryption Key (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;
- Miscellaneous control information – used to keep track of additional security relevant account attributes such as allowable periods of usage, whether the account has been locked, whether the password has expired, password history, and time since the password was last changed.
- Other non-security relevant information – used to complete the definition with other useful information such a user's real name and the purpose of the account.

The actual composition of the user attribute database depends upon the type of TSF (e.g., stand-alone, domain member, Domain Controller (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 Active Directory services (every TSF with the Active Directory 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 Active Directory that apply to selected TSFs (i.e., 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 Active Directory 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.

6.1.3.3.2 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).

6.1.3.4 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 Key Distributed Center (KDC) is available. Generally, each DC includes a KDC in addition to its Active Directory. If no KDC is available, NTLM will be used. In the evaluation 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 Active Directory (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).

6.1.3.5 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 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.

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.

6.1.3.6 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.

-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.

6.1.3.7 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 administrator guide recommends that the minimum password length be configured to no less than eight characters (with at least 90 available characters, the password space exceeds 4,300,000,000,000,000 available combinations). However, the minimum password length can be configured to require a password as large as 14 characters.

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. The TSF uses 56-bit or 168-bit DES keys. 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 security functional requirements:

- 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 (i.e., authentication data), privileges, and logon rights, 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 1 in 250,000,000,000,000.
- FIA_UAU.2: An authorized administrator can configure the TSF to allow no TSF-mediated functions prior to authentication .
- FIA_UAU.7: During an interactive logon, the TSF echoes the users password with "*" characters to prevent disclosure of the user's password.
- FIA_UID.2: An authorized administrator can configure the TSF to allow no TSF-mediated functions prior to identification .
- 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.
- BANNERS_EX, 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 their password has expired until their password has been changed.
- 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.
- 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.

6.1.4 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.

6.1.4.1 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.

6.1.4.2 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 (i.e., security descriptor), and occasionally by a specific SID requirement (i.e., “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 (create, delete, and 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 an authorized administrator to define constraints for passwords, 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.

Account Database – 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.

IP Security (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.

Encrypted File System (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 – 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.

SFR Mapping

The **Security Management function** satisfies the following security functional requirements:

- 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_MTD.1(a): Only an authorized administrator can create, delete, or clear the security event log.
- FMT_MTD.1(b): Only an authorized administrator can access the security event log.
- FMT_MTD.1(c): Only an authorized administrator can define user accounts and group accounts, define user/group associations (i.e., 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).
- 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.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_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 (i.e., passwords) by providing their old password for authorization.

6.1.5 TSF Protection Function

The TSF Protection provides:

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

6.1.5.1 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 (e.g., interrupt handling, memory management, task management, privileged instructions).

6.1.5.2 Internal TOE Protection

The TOE protects against unauthorized disclosure and modification of TSF data when it is transferred between physically separated parts of the TOE using a suite of Internet standard protocols including Internet Protocol Security (IPSec) and Internet Security Association and Key Management Protocol (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 directory service (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 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 Triple DES CBC. In addition to ESP, the TSF implements IP Authentication Header (AH). AH provides integrity, authentication and antireplay. AH uses a hashing algorithm, such as SHA-1, to compute a keyed message hash for each IP packet.

For outbound IP traffic, the processing occurs as follows:

- The IP stack calls into the IPSEC module to apply security action (encrypt, sign, etc.) 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 security association (SA).
- IPSEC action is performed (encrypt, sign, 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) and removes security specific headers.

6.1.5.3 TSF Data Replication Consistency

The Active Directory service allows for specific data to be replicated within the TOE. The Active Directory namespace includes a domain tree structure and a forest structure to facilitate the management of large size installations. Additionally, the Active Directory 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 global catalog. 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 2000 domains sharing a common schema, configuration, and global catalog, joined together to form a contiguous namespace. All domains in a given tree trust each other through transitive hierarchical Kerberos trust relationships. A larger tree can be constructed by joining additional domains as children to form a larger contiguous namespace.

A Global Catalog Server is a domain controller that stores specific information about all objects in the forest. The Global Catalog 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 domain controller 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 domain controllers and also to all Global Catalog Servers.

The major difference between a forest and a domain tree is the removal of the requirement that the member domains always form a tree.

Enterprises can be a single-tree or a multitree. 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 (GPO) and Directory Service (DS) data. Group Policy Objects are used to define configurations for groups of users and computers. Group Policy Objects 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 Group Policy Object 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 domain controller, those changes are replicated to the SYSVOL folders on all other domain controllers in the domain. SYSVOL content uses the same replication schedule as the DS for inter-site replication.

Along with the GPO, all domain controllers contain three types of DS data: domain, schema, and configuration. In the case of the global catalog server a forth category consisting of a partial replica of domain data for all domains is added. Each type of data is separated into distinct directory partitions that form the basic units of replication for the DS. These partitions are as follows:

- **Domain partition;** all objects in the directory for a given domain, 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 Active Directory, common to all domains in the domain tree or enterprise, and replicated to all domain controllers in the enterprise.
- **Configuration partition;** replication topology and related metadata, common to all domains in the domain tree or enterprise, replicated to all domain controllers in the enterprise.

Global catalog 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 domain controller (DC) in the enterprise. This extended model is referred to as Flexible Single Master Operation (FSMO). Currently in Windows 2000 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.
- RID (Relative Identifier) master - the single DC responsible for processing RID Pool requests from all DCs within a given domain.
- PDC emulator - a Windows 2000 DC that advertises itself as the primary domain controller (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 domain controller holds a table containing entries for its own USN and the USNs of its replication partners. During replication, the domain controller 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 domain controllers, the domain controllers 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 binaurally the same, and one can be discarded.

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

6.1.5.4 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 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).

6.1.5.5 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.

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.

6.1.5.6 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 (i.e., "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 domain controllers (DC) 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 security functional requirements:

- TRANSFER_PROT_EX – 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.

- REPLICATION_EX – The TSF provides consistency of replicated of Group Policy Objects and Directory Service data by implementing a well-defined TSF replication algorithm.

- FPT_RVM.1 – The TSF provides reference mediation primarily 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 check on all use of an object.

- 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 2000 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.

6.1.6 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 security functional requirement:

- 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 or group account can use on a specific NTFS disk volume.

6.1.7 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. 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 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.

SFR Mapping

The **Session Locking function** satisfies the following security functional requirement:

- FTA_SSL.1: Windows 2000 allows users 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 2000 also allows a user to directly invoke the session lock as described above.
- 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 2000 to satisfy the Common Criteria 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 measures applied by Microsoft ensure that configuration items 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 configuration item modifications are properly controlled. Microsoft performs configuration management on the TOE implementation representation, design, tests, user and administrator guidance, the CM documentation, and security flaws. These activities are documented in the Windows 2000 Configuration Management 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 2000 Configuration Management 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 2000 Life Cycle Management Plan.

6.2.1.3 Security Assurance Measure (SAM) Mapping

The Process Assurance measure satisfies the following Assurance requirements:

- 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 2000 into the evaluated configuration. These procedures are documented in the Windows 2000 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 2000 Administrator Guide

SAM Mapping

The Delivery and Guidance assurance measure satisfies the following Assurance requirements:

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

6.2.3 Design Documentation

The Windows 2000 "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 security assurance measure** satisfies the following security assurance requirement:

- 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_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_IMP.1: A subset of the source code used to generate the TOE satisfies this requirement.
- 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 2000 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 assurance requirements:

- 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 2000 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 2000 Administrator Guide

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

- The Windows 2000 Vulnerability Analysis

6.2.5.2 Strength of TOE Security Functions and Vulnerability Analysis

The strength of TOE security function analysis demonstrates that the SOF claims made in the ST for all probabilistic or permutation mechanisms are correct. Microsoft performs 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 resistant to obvious penetration attacks. The SOF and vulnerability analysis are documented in:

- The Windows 2000 Vulnerability Analysis

The Vulnerability Assessment assurance measure satisfies the following assurance requirements:

- 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.

7.1 Protection Profile Reference

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

7.2 PP 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.

7.3 Protection Profile 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, security functional requirements, and assurance requirements. The ST has primarily added additional items, or in the case of assurance requirements, enhanced requirements to those from EAL 3 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 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
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
SFR	FAU_GEN.1	Refinement
SFR	FAU_SEL.1	Removed
SFR	FCS_COP.1	Addition
SFR	FIA_AFL.1	Addition
SFR	FIA_SOS.1	Refinement
SFR	FIA_UAU.2	Addition
SFR	FIA_UID.2	Addition
SFR	FMT_MOF.1(a)	Addition
SFR	FMT_MOF.1(b)	Addition
SFR	FMT_MOF.1(c)	Addition
SFR	FMT_MSA.1(b)	Addition
SFR	FMT_MTD.1(e)	Addition
SFR	FMT_MTD.1(f)	Addition
SFR	FMT_MTD.1(g)	Addition
SFR	FMT_MTD.1(h)	Addition
SFR	FMT_MTD.1(i)	Addition
SFR	FMT_MTD.1(j)	Addition
SFR	FMT_MTD.1(k)	Addition
SFR	FMT_MTD.2	Addition
SFR	FMT_SAE.1	Addition
SFR	FMT_SMR.3	Addition
SFR	FPT_AMT.1	Removed
SFR	TRANSFER_PROT_EX	Addition
SFR	REPLICATION_EX	Addition
SFR	FRU_RSA.1	Addition
SFR	FTA_SSL.1	Addition
SFR	FTA_SSL.2	Addition
SFR	BANNERS_EX	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_DEL2	Upgrade for EAL4
SAR	ADV_FSP.2	Upgrade for EAL4
SAR	ADV_IMP.1	Addition 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	AVA_MSU.2	Upgrade for EAL4
SAR	AVA_VLA.2	Upgrade for EAL4

7.3.1 Protection Profile Differences

There are two requirements in the CAPP that are not included in this ST. They are FPT_AMT.1 and FAU_SEL.1. Even though these requirements are not included in this ST, this ST is still compliant with the CAPP. The rationale is provided below.

FPT_AMT.1 requires there be a suite of test available to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF. Typically, the term "underlying abstract machine" refers to the hardware components upon which the TSF has been implemented. This requirement is only applicable to a TOE if the TOE relies upon security assumptions of the underlying abstract machine. This ST does not include any assumptions on an abstract machine. The TOE described in this ST includes the hardware components. Therefore, this requirement is not applicable to this TOE. Given this, the objective the CAPP identifies is supported by FPT_AMT.1 (O.Enforcement) is totally satisfied in this ST by the TOE meeting FPT_RVM.1 and FPT_SEP.1. Additionally, there are no requirements in the CAPP that are dependent upon FPT_AMT.1.

FAU_SEL.1 (Selective Audit) and FAU_SAR.3 (Selective Audit Review) are both included in the CAPP to meet the objective O.Auditing and O.Manage. These requirements support these objectives by allowing for audit information to be presented to and managed by the authorized administrator. FAU_SEL.1 requires a pre-selection capability of events and FAU_SAR.3 requires a post-audit selection capability of events. To meet the CAPP objectives, each audit event type need not be selectable (based on the specified attributes) on both a pre-selection and post-selection basis but all event types must be selectable through some combination of these means. This ST meets the CAPP objectives by including one of the selective related requirements; FAU_SAR.3. The CAPP objectives, O.Auditing and O.Manage, are met in this ST by it requiring post-audit selection capability of "each" audit event type based on the attributes specified in the requirement. This requirement mandates the presentation of audit information to the authorized administrator and allows for the management of the audit information by the authorized administrator. Additionally, there are no requirements in the CAPP that are dependent upon FAU_SEL.1.

7.3.2 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.

7.3.3 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 polices 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.

7.3.4 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.

7.3.5 Security Functional Requirement Enhancements

The additional security functional requirements 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. FIA_UAU.2 and FIA_UID.2 are hierarchical to the CAPP requirements FIA_UAU.1 and FIA_UID.1. As indicated in Table 7-1 three requirements in the CAPP were further refined. These requirements are FAU_GEN.1, FDP_ACF.1, and FIA_SOS.1. 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 compliancy 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. FIA_SOS.1 is refined further than the CAPP to require a stronger secret than that specified in the CAPP. FDP_ACF.1 is refined further than the CAPP to add additional security attributes associated with a subject that the DAC policy is based upon. These refined SFRs are still compliant with the CAPP.

7.3.6 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 Security Objectives for IT Environment 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.UNAUTH_ACCESS T.SYSACC P.AUTHORIZED_USERS
O.DISCRETIONARY_ACCESS	T.USER_CORRUPT P.NEED_TO_KNOW
O.AUDITING	T.UNDETECTED_ACTIONS P.ACCOUNTABILITY
O.AUDIT_PROTECTION	T.AUDIT_CORRUPT
O.RESIDUAL_INFORMATION	P.NEED_TO_KNOW T.OBJECTS_NOT_CLEAN
O.MANAGE	P.ACCOUNTABILITY P.AUTHORIZED_USERS P.NEED_TO_KNOW
O.ENFORCEMENT	P.ACCOUNTABILITY

	P.AUTHORIZED_USERS P.NEED_TO_KNOW P-ADD-IPSEC
O.PROTECT	T.UNAUTH_ACCESS T.UNAUTH_MODIFICATION T.CONFIG_CORRUPT T.USER_CORRUPT
O.TRUSTED_PATH	T.SPOOF
O.LEGAL_WARNING	P.WARN
O.LIMIT_AUTHORIZATION	P.AUTHORIZATION
O.IPSEC	P-ADD-IPSEC
O.ENCRYPTED_DATA	T.USER_CORRUPT T.UNAUTH_ACCESS

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. Additionally, this objective implements the policy P.AUTHORIZED_USER by ensuring that only authorized users gain 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. 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.

O.AUDIT_PROTECTION – By ensuring that the audit information is protected, the threat T.AUDIT_CORRUPT is countered because unauthorized access will be prevented and audit information will not be lost.

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.

O.ENFORCEMENT – By ensuring that organizational policies are enforced, the policies P.ACCOUNTABILITY, P. AUTHORIZED_USERS 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 and T.UNAUTH_MODIFICATION 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.

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 if used in an unauthorized manner.

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.

O.ENRYPTED_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

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 Security Objectives for the Non-IT Objectives 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 users cooperate with guidance to meet the objectives of the TOE.

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
FAU_GEN.1			X										

Requirement	O.AUTHORIZATION	O.DISCRETIONARY_ACCES	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
FAU_GEN.2			X										
FAU_SAR.1			X			X							
FAU_SAR.2			X										
FAU_SAR.3			X			X							
FAU_STG.1			X	X									
FAU_STG.3			X			X							
FAU_STG.4			X	X		X							
FCS_COP.1													X
FDP_ACC.1		X											
FDP_ACF.1		X											
FDP_RIP.2					X								
Note1_EX					X								
FIA_AFL.1	X												
FIA_ATD.1	X	X									X		
FIA_SOS.1	X												
FIA_UAU.2	X												
FIA_UAU.7	X												
FIA_UID.2	X												
FIA_USB.1_EX		X	X										
FMT_MSA.1(a)		X				X							
FMT_MSA.1(b)		X				X							
FMT_MSA.3		X				X							
FMT_MTD.1(a)			X			X							
FMT_MTD.1(b)			X			X							
FMT_MTD.1(c)						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												
FMT_MTD.2	X					X							
FMT_MOF.1(a)						X							
FMT_MOF.1(b)						X						X	
FMT_MOF.1(c)	X					X							
FMT_REV.1(a)						X					X		
FMT_REV.1(b)		X											
FMT_SAE.1	X					X							
FMT_SMR.1						X					X		
FMT_SMR.3						X							

Requirement	O.AUTHORIZATION	O.DISCRETIONARY_ACCES	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
TRANSFER_PROT_EX								X				X	
FPT_RVM.1							X						
FPT_SEP.1							X	X					
FPT_STM.1			X										
REPLICATION_EX						X							
FRU_RSA.1	X												
FTA_SSL.1	X												
FTA_SSL.2	X												
BANNERS_EX										X			
FTA_TSE.1	X												
FTP_TRP.1									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.

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.2, FIA_UID.2 and FIA_UAU.7 require a user to be identified and authenticated before any other TSF-mediation action on their behalf is allowed and prevent the user requesting access from receiving insightful authentication feedback during the authentication.

FTA_SSL.1, FTA_SSL.2, 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 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.

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.1 and FDP_ACF.1 define the DAC Security Functional Policies (SFPs), 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 DAC SFP 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.3, FMT_MSA.1(b), and FMT_REV.1(b) restrict the ability to modify object security attributes to authorized users, ensures that restrictive default values are defined 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.ENCRYPTED_DATA

FCS_COP.1 prevents the decryption of encrypted data if the user attempting decryption is not the user that encrypted the data.

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.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, 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 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.,

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

FMT_MSA.1(a), FMT_MSA.3, FMT_MSA.1(b), 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(h), FMT_MTD.1(i), FMT_MTD.1(j), FMT_MTD.2, FMT_MOF.1(c), and FMT_SAE.1 ensure the authorized administrator can manage authentication data.

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

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.

REPLICATION_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 and FPT.SEP.1 ensure the TOE makes and enforces the decisions of the TSPs and that the TSF is protected from interference that would prevent if 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.

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.

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

BANNERS_EX 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; 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

TRANSFER_PROT_EX and FMT_MOF.1(b) ensure 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.

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.

8.2.2 Security Assurance Requirements 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. 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); 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 with ALC_FLR.3 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) is hierarchical to the EAL specified in the CAPP (EAL3). EAL 4 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.

> 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 BANNERS_EX and TRANSFER_PROT_EX, there are no CC identified dependencies for the CC requirements these explicit requirements are based upon (FTA_TAB.1 and FPT_ITT.1). Considering the changes between the CC requirements and the explicit requirements, the BANNERS_EX and TRANSFER_PROT_EX explicit requirements are dependent upon the TOE providing the functionality to allow the administrator to enable or disable the functionality described in these explicit requirements. Therefore, BANNERS_EX is dependent upon FMT_MTD.1.1(i) and TRANSFER_PROT_EX is dependent upon FMT_MOF.1.1(a).
>
> For REPLICATION_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 and FPT_ITT.1; and REPLICATION_EX and FPT_TRC.1).

For the FCS_COP.1 requirement, the CC identifies the following dependencies: FDP_ITC.1 or FCS_CKM.1, FCS_CKM.4, and FMT_MSA.2. The dependencies for this requirement are 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.
>
> FCS_CKM.1 and FCS_CKM.4: these requirements are concerned with key generation (FCS_CKM.1) and key destruction (FCS_CKM.4) and are applicable to cryptographic operations that rely upon the secure management of keys. The key used in the DES and Triple DES algorithm specified in FCS_COP.1 is used to encrypt and decrypt raw data and user private keys. The key is automatically created upon an encryption request and is automatically deleted upon decryption. Additionally, the key used to encrypt files (FEK) is automatically generated upon a request to encrypt a file and is automatically destroyed upon a request to delete a file. Therefore, there is no need to

specify a key generation algorithm and a key destruction method using FCS_CKM.1 and FCS_CKM.4.

FMT_MSA.2: this requirement is concerned with ensuring that only secure values are accepted for security attributes. There are no security attributes entered within the context of the operations specified by FCS_COP.1, therefore, FMT_MSA.2 is not applicable.

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	48
2	FAU_GEN.2	FAU_GEN.1 FIA_UID.1	1, 19
3	FAU_SAR.1	FAU_GEN.1	1
4	FAU_SAR.2	FAU_SAR.1	3
5	FAU_SAR.3	FAU_SAR.1	3
6	FAU_STG.1	FAU_GEN.1	1
7	FAU_STG.3	FAU_STG.1	6
8	FAU_STG.4	FAU_STG.1	6
9	FCS_COP.1	FDP_ITC.1 or FCS_CKM.1 FCS_CKM.4 FMT_MSA.2	N/A
10	FDP_ACC.1	FDP_ACF.1	11
11	FDP_ACF.1	FDP_ACC.1 FMT_MSA.3	10, 26
12	FDP_RIP.2	None	
13	Note1_EX	None	
14	FIA_AFL.1	FIA_UAU.1	17
15	FIA_ATD.1	None	
16	FIA_SOS.1	None	

Component No.	ST Functional Requirement		Satisfied Component No.
17	FIA_UAU.2	FIA_UID.1	19
18	FIA_UAU.7	FIA_UAU.1	17
19	FIA_UID.2	None	
20	FIA_USB.1_EX	FIA_ATD.1	15
21	FMT_MOF.1(a)	FMT_SMR.1	42
22	FMT_MOF.1(b)	FMT_SMR.1	42
23	FMT_MOF.1(c)	FMT_SMR.1	42
24	FMT_MSA.1(a)	FDP_ACC.1 or FDP_IFC.1, FMT_SMR.1	10, 42
25	FMT_MSA.1(b)	FDP_ACC.1 or FDP_IFC.1, FMT_SMR.1	10, 42
26	FMT_MSA.3	FMT_MSA.1(a) FMT_SMR.1	24, 42
27	FMT_MTD.1(a)	FMT_SMR.1	42
28	FMT_MTD.1(b)	FMT_SMR.1	42
29	FMT_MTD.1(c)	FMT_SMR.1	42
30	FMT_MTD.1(d)	FMT_SMR.1	42
31	FMT_MTD.1(e)	FMT_SMR.1	42
32	FMT_MTD.1(f)	FMT_SMR.1	42
33	FMT_MTD.1(g)	FMT_SMR.1	42
34	FMT_MTD.1(h)	FMT_SMR.1	42
35	FMT_MTD.1(i)	FMT_SMR.1	42
36	FMT_MTD.1(j)	FMT_SMR.1	42
37	FMT_MTD.1(k)	FMT_SMR.1	42
38	FMT_MTD.2	FMT_MTD.1(d) FMT_SMR.1	30, 42
39	FMT_REV.1(a)	FMT_SMR.1	42
40	FMT_REV.1(b)	FMT_SMR.1	42
41	FMT_SAE.1	FMT_SMR.1 FPT_STM.1	42, 50
42	FMT_SMR.1	FIA_UID.1	19
43	FMT_SMR.3	FMT_SMR.1	42
44	TRANSFER_PROT_EX	FMT_MTD.1.1(b)	22
45	REPLICATION_	TRANSFER_PROT_EX	44

Component No.	ST Functional Requirement		Satisfied Component No.
	EX		
46	FPT_RVM.1	None	
47	FPT_SEP.1	None	
48	FPT_STM.1	None	
49	FRU_RSA.1	None	
50	FTA_SSL.1	FIA_UAU.1	17
51	FTA_SSL.2	FIA_UAU.1	35
52	BANNERS_EX	FMT_MTD.1.1(i)	
53	FTA_TSE.1	None	
54	FTP_TRP.1	None	

8.2.4 Explicitly Stated Requirements Rationale

The ST includes explicitly stated requirements: Note1_EX; FIA_USB.1_EX; REPLICATION_EX; TRANSFER_PROT_EX; BANNERS_EX. Note1_EX and FIA_USB.1_EX, referred to as FDA_RIP.2.Note1 and FIA_USB.1 in the CAPP, are included in the CAPP along with a rationale for each requirement.

REPLICATION_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 REPLICATION_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, REPLICATION_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. REPLICATION_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.

TRANSFER_PROT_EX:

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, the ST authors created the explicit requirement TRANSFER_PROT_EX. The functionality to "always" protect TSF 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 objective O.PROTECT. This change allows the authorized administrator to be able to disable this functionality and remain within the evaluated configuration.

BANNERS_EX:

To address the TOE functionality of displaying an advisory warning message before logon to support the objective O.LEGAL_WARNING, the FTA_TAB.1 CC requirement was considered. However, because

FTA_TAB.1 prescribes functionality beyond what is required to meet O.LEGAL_WARNING, the ST authors created the explicit requirement BANNERS_EX. O.LEGAL_WARNING specifies that a mechanism be available be provided to advise users of legal issues before they are allowed access to resources. It does not state that the message always be displayed. FTA_TAB.1 requires that the TSF display an advisory warning message regarding unauthorized use of the TOE. The ST authors added the words "be able to" to the requirement to provide the desired flexibility in the evaluated configuration to meet the objective O.LEGAL_WARNING. This change allows the authorized administrator to disable this functionality and remain within the evaluated configuration.

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 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 Strength of Function Rationale

The TOE minimum strength of function of SOF-medium was chosen to be consistent with the CAPP. The explicit strength of function 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 TOE Summary Specification Rationale

This Section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions are suitable to meet the TOE security requirements and the assurance measures address the assurance measures. The collection of security functions 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 security assurance requirements as indicated in Table 8-6. The security functions and assurance measures described in the TOE summary specification 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	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	X						
FAU_STG.1	X						
FAU_STG.3	X						
FAU_STG.4	X						
FCS_COP.1		X					
FDP_ACC.1		X					
FDP_ACF.1		X					
FDP_RIP.2		X					
Note1_EX		X					
FIA_AFL.1			X				
FIA_ATD.1			X				
FIA_SOS.1			X				
FIA_UAU.2			X				
FIA_UAU.7			X				
FIA_UID.2			X				
FIA_USB.1_EX			X				
FMT_MSA.1(a)		X					
FMT_MSA.1(b)		X					
FMT_MSA.3		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.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			

Requirement	AUDIT	USER DATA PROTECTION	I & A	SECURITY MANAGEMENT	TSF PROTECTION	RESOURCE UTILIZATION	SESSION LOCKING
FMT_SMR.3			X				
TRANSFER_PROT_EX					X		
FPT_RVM.1					X		
FPT_SEP.1					X		
FPT_STM.1					X		
REPLICATION_EX					X		
FRU_RSA.1						X	
FTA_SSL.1							X
FTA_SSL.2							X
BANNERS_EX			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		

Requirement	PROCESS ASSURANCE	DELIVERY AND GUIDANCE	DESIGN DOCUMENTATION	TEST	VULNERABILITY ASSESSMENT
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

APPENDIX A List of Acronyms

ACE	Access Control Entry
ACL	Access Control List
ACM	Access Control Management
ACP	Access Control Policy
AGD	Administrator Guidance Document
AH	Authentication Header
CAPP	Controlled Access Protection Profile
CBC	Cipher Block Chaining
CD-ROM	Compact Disk Read Only Memory
CM	Control Management
COM	Common Object Model
CPU	Central Processing Unit
DAC	Discretionary Access Control
DACL	Discretionary Access Control List
DC	Domain Controller
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DFS	Distributed File System
DNS	Domain Name System
DO	Delivery Operation
DS	Directory Service
EAL	Evaluation Assurance Level
EFS	Encrypting File System
ESP	Encapsulating Security Protocol
FEK	File Encryption Key
FSMO	Flexible Single Master Operation
GB	Gigabyte
GPC	Group Policy Container
GPO	Group Policy Object
GPT	Group Policy Template
GUID	Globally Unique Identifiers
HMAC	Hash-Based Message Authentication Code
I/O	Input/Output
IP	Internet Protocol
IPC	Interprocess Communication

IPSec	Internet Protocol Security
KDC	Key Distributed Center
LDAP	Lightweight Directory Access Protocol
LSA	Local Security Authority
LPC	Local Procedure Call
LSA	Local Security Authority
MAC	Message Authentication Code
MBR	Master Boot Record
NTFS	NT File System
NTLM	NT LAN Manager
PAE	Physical Address Extension
RID	Relative Identifier
RPC	Remote Procedure Call
SA	Security Association
SACL	System Access Control List
SAM	Security Assurance Measure
SD	Security Descriptor
SID	Security Identifier
SF	Security Functions
SFR	Security Functional Requirements
SRM	Security Reference Monitor
ST	Security Target
TOE	Target of Evaluation
TSF	Target of Evaluation Security Functions
UNC	Universal Naming Convention
USB	Universal Serial Bus
USN	Update Sequence Number
X86	Intel Microprocessors

Appendix B TOE Component Decomposition

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
1.	Active Directory Replication Management	It permits a client to manage the replication topology by adding, modifying, and deleting directory server references from the topology, retrieve information about the replication topology, verify the consistency of the replication topology for an Active Directory site, and start replication for a naming context on a directory server.	ntdsa.dll (lsass.exe)	RPC
2.	Core Directory Service	It is the repository in which domain-wide resources are managed and shared.	ntdsa.dll (lsass.exe)	N/A
3.	DS Role Management	It is a collection of RPC-based system APIs for managing a role of a server within a domain.	ntdsetup.dll (lsass.exe)	RPC
4.	Encrypting File System Management Service	It provides a key management mechanism for users to encrypt and decrypt files.	lsasrv.dll (lsass.exe)	RPC
5.	Inter-Site Messaging	It is a service that supports synchronous and asynchronous replication of directory information between sites.	ismserv.exe ismsmtp.dll ismsink.dll	RPC
6.	IP Security Policy Agent	It is the service within the IPSec architecture that provides the Administrator's interface for creating and maintaining the IPSec Policy at both the Domain and Local level.	polagent.dll (lsass.exe)	RPC
7.	Kerberos Key Distribution Center (KDC)	The Key Distribution Center provides server-side services for the Kerberos subprotocols Authentication Service (AS) and Ticket Granting Service (TGS). The KDC also supports a change password request interface.	kdcsvc.dll (lsass.exe)	Network protocol
8.	Kerberos Authentication / Security	As an authentication package, Kerberos performs user authentication for LSA	Kerberos.dll (lsass.exe)	LPC

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
	Package	Authentication in support of the LsaLogonUser() API. As a security package, it implements a network security protocol, in this case Kerberos version 5, that supports remote user authentication and other network security services.		
9.	LDAP Server	LDAP is the primary generic interface to the Core DS services. The interface to the LDAP server is via TCP/UDP port number 389. If SSL is used, the interface to the LDAP server is via port number 636.	ntdsa.dll (lsass.exe)	Network protocol
10.	LSA Audit	It is responsible for receiving all security audit messages and forwarding the audit records contained therein, to the Event Logger subcomponent for inclusion in the security event log.	lsasrv.dll (lsass.exe)	LPC
11.	LSA Authentication	It provides services to coordinate, manage, and control the logon authentication process using various authentication packages.	lsasrv.dll (lsass.exe)	LPC
12.	LSA Policy	It provides the programming interfaces to manage, configure, and set most locally-focused aspects of a system's security policy. These policies include the audit policy, assignment of privilege and logon rights, and definition of trust relationships.	lsasrv.dll (lsass.exe)	RPC
13.	MAPI Based Directory Request	It is a collection of RPC-based system APIs that allows messaging applications to access the core directory services.	ntdsa.dll (lsass.exe)	RPC
14.	MSV1_0 Authentication / Security Package	As an authentication package, It performs user authentication for LSA Authentication in support of the LsaLogonUser() API. As a security package, it	msv1_0.dll (lsass.exe)	LPC

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
		implements a network security protocol, in this case NTLM, that supports remote user authentication and other network security services.		
15.	Netlogon	It provides functionality that makes a network of cooperating machines appear as a single system in terms of authentication. Generally, this is accomplished by replicating the authentication data and distributing that data across the network.	netlogon.dll (lsass.exe)	RPC and Mailslot
16.	DS Backup Restore	It allows a domain controller to backup the Directory database to a set of files, and to restore these files to the Directory database.	ntdsbsrv.dll (lsass.exe)	RPC
17.	ISAKMP/Oakley Protocol Processing Support for IPSEC	It is statically linked into the user-mode Policy Agent service. It provides the Policy Agent service with routines to implement the Oakley protocol and the ISAKMP protocol.	oakley.dll (lsass.exe)	N/A
18.	Security Accounts Manager (SAM)	It provides a set of interfaces for creating, protecting, and manipulating a set of system objects that represents user, group, and machine accounts, as well as associated system-wide policies. The SAM interfaces are legacy interfaces in that they provide backwards compatibility with previous versions of the operating system.	samsrv.dll (lsass.exe)	RPC
19.	Schannel Security Package	It is a security support provider (SSP) that provides support for four network security protocols, namely Secure Sockets Layer (SSL) versions 2.0 and 3.0, Transport Layer Security (TLS) version 1.0 and Private Communications Technology (PCT) version 1.0.	schannel.dll (lsass.exe)	LPC

IO Component Decomposition

Subcomponents of IO belong to one of the following categories: IO Core, IO File, IO Net, and IO Device.

IO Core Component Decomposition

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
20.	IO Manager	It is a framework within which all kernel-mode drivers controlling and interfacing with peripheral devices reside.	ntoskrnl.exe, ntdll.dll	User / Kernel mode transition via system service table
21.	File System Recognizer	It implements a surrogate driver that performs preliminary file system recognition. FSRec minimizes memory usage, because it avoids loading all file system drivers, regardless of whether or not they have any volumes to manage.	fs_rec.sys	N/A
22.	Mount Manager	It assigns drive letters and mount points for all physical disk volumes in the system (including basic and dynamic disk volumes).	mountmgr.sys	IOCTLs

IO File Component Decomposition

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
23.	CDFS	It supports PC CDROMs and provides a read only format, as defined by the International Organization for Standardization (ISO) computer standard 9660, level 2.	cdfs.sys	FSCTLs, and IRPs
24.	EFS Driver	It is a kernel-mode driver that implements cryptographic protection of files and directories on NTFS volumes.	efs.sys	N/A
25.	Fast FAT	It consists of a single kernel-mode file system driver that implements the File Allocation Table (FAT) file system.	fastfat.sys	FSCTLs, and IRPs
26.	Mailslot	It provides a connectionless “unreliable” mechanism that demonstrates a high rate of delivery in practical application.	mailslpt.sys	FSCTLs, and IRPs
27.	NamedPipe	It provides a connection-oriented protocol, based on Server Message Blocks	npfs.sys	FSCTLs, and IRPs

		(SMBs) and Network Basic Input/Output System (NetBIOS), that is a reliable bidirectional means of synchronous or asynchronous communication between a server process and 1 or more client processes.		
28.	NTFS	This file system supports object-oriented applications by implementing files as objects with user- and system-defined attributes. NTFS includes security descriptor based access control to files and directories, multiple data streams per file, file system recovery, encryption, compression, disk quotas, national locale Unicode file names, and other advanced features.	ntfs.sys	FSCTLs, and IRPs

IO Net Component Decomposition

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
29.	Ancillary Function Driver (AFD)	It is a kernel-mode driver that provides support for Windows Sockets to communicate with underlying transports.	afd.sys	IOCTLs and IRPs
30.	Browser	It is a kernel-mode driver that functions as a datagram receiver for broadcast massages such as server announcements, browser updates, domain accounts database synchronization, and small mailslot writes. It is a part of the I/O component and works closely with the Network Redirector, Netlogon, and Browser service subcomponents.	mrxsmb.sys	IOCTLs and IRPs
31.	DFS Server Driver	It provides functionality to unite files on different computers in a single logical namespace. It allows multiple file servers and file server shares to be represented as a unified hierarchy.	dfs.sys	IOCTLs
32.	IP Filter Driver	It is a kernel-mode driver that cooperates with the TCPIP	ipfltdrv.sys	IOCTLs and IRPs

		subcomponent to provide filtering of IP packets based on a set of filtering rules. The IP Filter Driver can also be used to install a single filter-hook driver.		
33.	IP in IP Tunnel Driver	It is a kernel-mode driver that provides a capability to encapsulate IP traffic into the payload of another IP packet. Thus it creates a tunnel using IP packet in an IP packet (i.e., an IP-in-IP tunnel).	ipinip.sys	IOCTLs
34.	IPSEC Driver	It cooperates with user mode and kernel mode subcomponents to implement the IPSec protocol as described by the Network IO and/or Security component documents. The IPSec driver is responsible for managing IPSec kernel data structures and converting IP packets into IPSec packets (and back).	ipsec.sys	IOCTLs, IRPs, and network protocols
35.	MS Generic Packet Classifier	It provides a means by which packets internal to a specific network node can be classified, and consequently prioritized to provide a desired handling of packets.	msgpc.sys	IOCTLs
36.	Multiple UNC Provider and DFS Client	It locates the provider of a resource on a network using the UNC name of the resource. The MUP finds the appropriate redirector for all UNC name-based I/O requests, including files, Named Pipes, and Mailslots.	mup.sys	FSCTLs, and IRPs
37.	Network Driver Interface Specification (NDIS)	NDIS abstracts the network hardware from network drivers (e.g., tdi.sys, tcpip.sys, ipsec.sys).	ndis.sys	N/A
38.	Net Detect	It supports the detection of network interface cards.	netdet.sys	IOCTLs
39.	NetBIOS over TCP/IP (NetBT)	It implements NetBIOS services over the TCP/IP protocol.	netbt.sys	IOCTLs, and IRPs
40.	Packet Scheduler	It supports the traffic control aspect of the operating system’s quality of service (QoS) network communication features.	psched.sys	N/A
41.	Redirector	It presents itself as a file system and provides the	rdbss.sys, mrxsmb.sys	FSCTLs and IOCTLs, and

		client-side interface to access remote resources. This allows users to request access to remote data just as they would request data from the local file system.		IRPs
42.	Server Driver	It responds to network connection and service requests from the Redirector (rdr.sys) of a remote computer system. The Server Driver also responds to the Server Service (service.exe) to allow configuration and management through Network Management APIs (netapi32.dll).	srv.sys	FSCTLs, IRPs, and network protocol (SMB)
43.	Transmission Control Protocol / Internet Protocol (TCPIP)	It is the TCPIP stack in the kernel mode.	tcpip.sys	IOCTLs, and IRPs
44.	Transport Driver Interface (TDI)	The TDI interface is exposed by all Windows NT transport drivers. The TDI interface specification describes the set of primitive functions by which transport drivers and TDI clients communicate, and the call mechanisms used for accessing them.	tdi.sys	IOCTLs, and IRPs

IO Device Component Decomposition

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
45.	3COM el90xbc5 NIC Miniport Driver	It allows the system to communicate with 3COM Ethernet PCI adapters or network interface cards (NICs), and provides support for Plug and Play (PnP), Power Manager, and Windows Management Instrumentation (WMI).	el90xbc5.sys	N/A
46.	Accelerated Graphics Port (AGP) Bus Filter Driver	It is a high-speed interface between the system chipset and the graphics controller providing enhanced graphics performance for 3D and video applications. AGP removes 3D and video traffic from the PCI bus and provides the ability to set aside system	agp440.sys	N/A

		memory for use by the graphics controller.		
47.	Advanced Configuration Program Interface (ACPI)	It handles temperature control, button control, Plug and Play (PnP), Power Manager, and Windows Management Instrumentation (WMI) operations for ACPI compatible devices. It enables operating system-directed power management and system configuration.	acpi.sys	IOCTLs, and IRPs
48.	ATI ATI2MPAB Graphics Miniport Driver	It manages information about the ATI graphics hardware. It works in combination with the display driver to provide graphic services.	atimpab.sys	N/A
49.	ATI ATIRAGE Graphics Miniport Driver	It manages information about the ATI graphics hardware. It works in combination with the display driver to provide graphic services.	atiragem.sys	N/A
50.	Adaptec adpu160m SCSI/RAID Controller	It is a kernel-mode mini-port driver for the SCSI port driver to provide the system the hardware specific functionality associated with h/w device that it supports.	adpu160m.sys	N/A
51.	Adaptec perc2 SCSI-RAID Controller and its perc2hib lower filter driver	It is a kernel-mode mini-port driver for the SCSI port driver to provide the system the hardware specific functionality associated with h/w device that it supports. Its perc2hib lower filter driver provides additional plug and play support.	perc2.sys, perc2hib.sys	N/A
52.	AMI mraid2k SCSI/RAID Controller	It is a kernel-mode mini-port driver for the SCSI port driver to provide the system the hardware specific functionality associated with h/w device that it supports.	mraid2k.sys	N/A
53.	Beep	It controls the speaker port to generate sounds. It allows tones of a specific frequency and duration to be generated.	beep.sys	IOCTLs
54.	Compact Disk Read Only Memory (CDROM)	It uses services provided by the Class Plug and Play (PnP) library (classpnp.sys), which Changer, Tape, and Disk storage class drivers also use.	cdrom.sys	IOCTLs
55.	Compaq cpq32fs2 SCSI	It is a kernel-mode mini-port driver for the SCSI port driver	cpq32fs2.sys	N/A

	Controller	to provide the system the hardware specific functionality associated with h/w device that it supports.		
56.	Compaq n100nt5 NIC Miniport Driver	It allows the system to communicate with Compaq Ethernet PCI adapters or network interface cards (NICs), and provides support for Plug and Play (PnP), Power Manager, and Windows Management Instrumentation (WMI).	n100nt5.sys	N/A
57.	Disk	It is a kernel-mode Small Computer Standard Interface (SCSI) hard disk class driver. It enumerates the Physical Device Objects (PDO) created by the Integrated Device Electronics (IDE) or SCSI subcomponents and creates an object directory to contain the objects for the hard disk and all its partitions.	disk.sys	IOCTLs, and IRPs
58.	Disk Performance Driver	It generates performance related statistics on raw disk access. It layers above the physical device objects that represent physical disks (storage devices), and supports Plug and Play (PnP), Power Manager, and Windows Management Instrumentation (WMI).	diskperf.sys	IOCTLs
59.	Federal Information Processing Standard (FIPS) Driver	It provides kernel mode callers cryptographic services.	fips.sys	N/A
60.	Floppy Disk	It consists of the file device controller driver (fdc.sys) and the floppy driver (floppy.sys). Together, the fdc driver and the floppy driver tell the floppy disk device where to put the drive head and whether to read or write.	fdc.sys, floppy.sys	IOCTLs
61.	I8042 Port Driver	It is a hardware dependent port driver for PS/2 style keyboard and mouse devices.	i8042prt.sys	IOCTLs
62.	Integrated Device Electronics (IDE)	It provides access to the devices attached to the Integrated Device Electronics (IDE) bus.	atapi.sys, pciidex.sys, pciide.sys, intelpci.sys	IOCTLs, and IRPs
63.	Intel e100bnt5	It allows the system to	e100bnt5.sys	N/A

	NIC Miniport Driver	communicate with Intel Ethernet PCI adapters or network interface cards (NICs), and provides support for Plug and Play (PnP), Power Manager, and Windows Management Instrumentation (WMI).		
64.	Keyboard	It provides generic and hardware independent operation of keyboard devices, and kbfiltr.sys, a filter driver.	kbdclass.sys, kbfiltr.sys	IOCTLs, and IRPs
65.	Kernel Security Device Driver (KsecDD)	It exports functions for other drivers that need to send Local Procedure Call (LPC) messages to the Local Security Authority Server (LSASRV).	ksecdd.sys	IOCTLs
66.	Logical Disk Manager	It oversees dynamic disks, allows the hard disk to be divided into partitions and supports multi-partition volumes.	dmload.sys, dmboot.sys, dmio.sys	IOCTLs
67.	Microcode Update Device	It is a kernel-mode device driver that applies the computer's central processing unit (CPU) microcode software update patches to processors on the running system.	update.sys	N/A
68.	MegaRAID35x SCSI/RAID Controller	It is a kernel-mode mini-port driver for the SCSI port driver to provide the system the hardware specific functionality associated with h/w device that it supports.	mraid35x.sys	N/A
69.	Mouse	It provides generic and hardware independent operation of mouse devices, and moufiltr.sys, a filter driver.	mouclass.sys, moufiltr.sys	IOCTLs and IRPs
70.	Null Driver	It is used as a place holder for a driver that does not exist, as a destination that makes output invisible, or as a way to prevent an unused device from being started.	null.sys	N/A
71.	NVIDIA NV4 Windows Graphic Miniport Driver	It manages information about the NVIDIA graphics hardware. It works in combination with the display driver to provide graphic services.	nv4.sys	N/A
72.	Parallel Port	It implements the processing	parport.sys,	N/A

	Driver	of I/O requests specific to a parallel port.		
73.	Parallel Class Driver	It provides the ability for multiple parallel devices to share a parallel port.	parallel.sys	N/A
74.	Partition Manager	It handles Plug and Play (PnP), Power Manager, and Windows Management Instrumentation (WMI) operations for disk partitions.	partmgr.sys	IOCTLs
75.	Plug and Play PCI Enumerator	It supports Plug and Play (PnP), Power Manager, and Windows Management Instrumentation (WMI) for PCI compatible devices.	pci.sys	N/A
76.	PnP Software Device Enumerator	It is a kernel-mode bus driver that provides access to the registered device driver interface classes for plug-and-play devices enumerated on the bus.	swenum.sys	IOCTLs and IRPs
77.	Redbook - Rendering of CD digital audio	It is a kernel mode system driver that manages the rendering of CD digital audio.	redbook.sys	IOCTLs and IRPs
78.	Serial Enumerator Driver	It is a kernel-mode driver used by the Serial Port Driver to enumerate devices on an RS-232 port.	serenum.sys	IOCTLs and IRPs
79.	Serial Port	It implements the processing of I/O requests specific to a serial port. It is a function driver for legacy COM ports and Plug and Play COM ports. It is also a lower-level device filter driver for Plug and Play devices.	serial.sys	IOCTLs and IRPs
80.	Small Computer System Interface (SCSI) Port	It supplies the system support needed to carry out I/O operations on SCSI devices.	scsiport.sys	IOCTLs and IRPs
81.	Video Graphics Adapter (VGA)	It manages information about the VGA hardware.	vga.sys	N/A
82.	Video Port	It manages the access to video devices.	videoprt.sys	IOCTLs and IRPs

Executive & Primitive Kernel Component Decomposition

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
83.	Cache Manager	It provides a high-speed, intelligent mechanism for reducing disk I/O and increasing overall system throughput.	ntoskrnl.exe	N/A
84.	Configuration	It implements the Registry,	ntoskrnl.exe	User / Kernel

	Manager	which is used by nearly all components of the system to store configuration data. The Registry is a hierarchical database whose elements are called keys. Keys are an NT object; each key has its own security descriptor.		mode transition via system service table
85.	Executive Object Services	It is a collection of system APIs for managing a broad range of system objects, primitives, and settings.	ntoskrnl.exe	User / Kernel mode transition via system service table
86.	Graphics Device Interface (GDI)	It is responsible for displaying graphics on video displays and printers. GDI is used extensively by applications, and by the operating system itself for generating user interface elements such as icons, window borders, menus, scroll bars, and cursors.	win32k.sys	User / Kernel mode transition via GDI specific system service table
87.	Hardware Abstraction Layer (HAL)	It is the software within kernel-mode responsible for abstracting hardware implementation details from the rest of the system to improve portability. Kernel-mode software components depend on the HAL, to accomplish tasks that are requested by other kernel-mode components and user-mode software applications.	hal.dll	N/A
88.	Kernel Runtime	It provides information about the build number of the operating system, what global flags are set and what the current thread execution block is.	ntoskrnl.exe	N/A
89.	Local Process Communication (LPC)	It implements inter-process communication between two threads in different processes on the same system.	ntoskrnl.exe	User / Kernel mode transition via system service table
90.	Memory Manager	It provides capabilities that facilitate the allocation, use, management, and de-allocation of memory.	ntoskrnl.exe	User / Kernel mode transition via system service table
91.	Micro Kernel	It works very closely with the HAL subcomponent to provide fundamental, low-level mechanisms used by higher level subcomponents to provide more sophisticated operating system services. It	ntoskrnl.exe	N/A

		is never paged out of memory cannot be preempted. It performs thread scheduling and context switching, exception and interrupt handling, low-level multiprocessor synchronization and processor specific functions including flushing the Translation Lookaside Buffer (TLB) and initialization.		
92.	Object Manager	It is responsible for creating, deleting, protecting, and tracking objects.	ntoskrnl.exe	User / Kernel mode transition via system service table
93.	Plug and Play (PnP) Manager	It is responsible for providing the operating system with the ability to recognize and adapt to hardware configurations.	ntoskrnl.exe	User / Kernel mode transition via system service table
94.	Power Manager	It implements system power policy.	ntoskrnl.exe	User / Kernel mode transition via system service table
95.	Process Manager	It provides user-mode interfaces for managing program execution within the OS.	ntoskrnl.exe	User / Kernel mode transition via system service table
96.	Security Reference Monitor (SRM)	It performs access validation on objects, tests for privileges, and generates audit events.	ntoskrnl.exe	User / Kernel mode transition via system service table
97.	Window Manager (User)	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.	win32k.sys	User / Kernel mode transition via User specific system service table
98.	Virtual DOS Machine (VDM)	It provides a virtual Intel 80486 computer environment in which applications written for legacy operating systems (16-bit MS-DOS and 16-bit Windows) can run.	ntoskrnl.exe, ntvdm.exe	User / Kernel mode transition via system service table
99.	Windows Management Instrumentation (WMI) Driver	The driver is instantiated by the I/O Manager during the I/O initialization process, and provides the implementation for WMI objects.	ntoskrnl.exe, wmilib.sys	IOCTLs and IRPs

Winlogon Component Decomposition

	Subcomponent	Description	Binary Location	Type of external

				Interfaces implementing Security Checks
100.	WinLogon/GINA	They control access to the TOE desktop. WinLogon and GINA work together to handle interactive user logons, user logoffs, shutdowns, and screen saver management.	winlogon.exe, msgina.dll	Graphical User Interfaces, and RPC
101.	Group Policy	It is implemented as an executable (userinit.exe), started by the Winlogon/GINA subcomponent. It is used to start the user's shell and perform other initialization when a new user logs on to the system. The Group Policy subcomponent also runs any Group Policy Object (GPO) scripts that are stored on the system.	userinit.exe	N/A
102.	Profile Mapping	It is used to map user profiles from the current profile owner's SID to a different SID.	profmap.dll (winlogon.exe)	RPC
103.	Syskey	It provides services related to the protection of the SAM user database, which stores user authentication information.	winlogon.exe	LPC
104.	User Environment	It provides capabilities related to Group Policy, User Profiles, and Environment Variables to the WinLogon subcomponent.	userenv.dll (winlogon.exe)	N/A
105.	Windows File Protection (WFP)	It is a collection of RPC-based system APIs that protect system files from being modified or deleted by unauthorized users.	sfc.dll (Winlogon.exe), wintrust.dll, mscat32.dll	RPC

Win32 Component Decomposition

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
106.	Client-Server Runtime SubSystem (CSRSS)	It is the user-mode protected server that supports the console, shutdown, and miscellaneous system functions (such as supporting portions of the VDM processes and hard error handling).	csrss.exe	LPC
107.	Base Server	Base Server functionality	basesrv.dll	LPC

		supports VDM processes, provides National Language Support and provides miscellaneous capabilities offered by the CRSSS process.	(csrss.exe)	
108.	Windows (Console/User) Server	It allows manipulation of consoles in the client's current window station and desktop. It offers window specific configuration routines used in support of Winlogon, Task manager, and general user requests.	winsrv.dll (csrss.exe)	LPC

Services Component Decomposition

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
109.	Alerter Service	It allows alert messages to be generated when a specified event occurs.	alrsvc.dll (services)	N/A
110.	Computer Browser	It is responsible for providing functions that assist the Browser service on specific servers.	browser.dll (services.exe)	RPC
111.	DHCP Service (Both Server and client)	They enable individual DHCP Clients on an IP network to extract network configuration data from a DHCP server.	dhcpcsvc.dll (services.exe), dhcpssvc.dll (services.exe), dhcpsapi.dll	RPC, and NamedPipe,
112.	Protection Storage (DPAPI) and Misc. certificate service	It provides storage for sensitive data that must be kept secret or free from modification. This protection prevents access by unauthorized users to another user's private data.	cryptsvc.dll, psbase.dll (services.exe)	RPC
113.	Event Logger	It is the service that records, stores, protects, and manages event logs.	eventlog.dll (services.exe)	RPC
114.	File Replication Service	It replicates system policies and logon scripts stored in a System Volume (SYSVOL).	ntfrs.exe	RPC
115.	TCP/IP NetBIOS Service	It issues I/O control codes to the NetBT kernel-mode driver and responds to events to accept requests for action from the NetBT kernel-mode driver.	lmhsvc.dll (services.exe)	N/A
116.	Logical Disk Manager (LDM) Watchdog	It is responsible for starting and stopping the LDM Administrative Service and for	dmserver.dll (services.exe)	N/A

	Service	listening for plug and play event notifications related to disk management.		
117.	Messenger Service	It receives messages and displays them on to the local user's desktop.	msgsvc.dll (services.exe)	N/A
118.	Plug and Play (PnP) Service	It manages devices and their drivers, giving the operating system the ability to recognize and adapt to hardware configurations with minimal user intervention.	umpnpmgr.dll (services.exe)	RPC
119.	Remote Registry Service	It accepts requests from remote clients to manipulate the registry database.	regsvc.exe	RPC
120.	Secondary Logon (aka RunAs) Service	It allows a user to start a new process running on behalf of a different user account.	seclogon.dll (services.exe)	NamedPipe
121.	Security Configuration Editor Engine Service	It responds to client side commands issued by the Security Configuration and Analysis, and the Security Templates Microsoft Management Console snap-ins.	scesrv.dll (services.exe)	RPC
122.	Server Service	It is manages shared network resources (shares), controls Server Driver, creates and deletes shares, sends periodic server announcements, and provides support for sharing resources and printing via the Server Message Block (SMB) protocol.	srvsvc.dll, xactsrv.dll (services.exe)	LPC, and RPC
123.	Service Control Manager	It is a system server that is started during system initialization and is used to create, start, and manage services and drivers on the local or a remote machine.	services.exe	RPC
124.	System Event Notification Service (SENS)	It works with the COM+ Event System. SENS is an event publisher for the classes of events that it monitors: network, logon, and power/battery events.	sens.dll (services.exe)	N/A
125.	Time Synchronization Service	It provides a mechanism for the TOE to use a common time source.	w32time.dll (services.exe)	RPC, and network protocol
126.	WMI Core Service	It marshals events that occur in the WMI kernel mode subcomponent and then dispatches them to the appropriate user-mode	wmicore.dll (services.exe)	RPC

		Management Applications. It handles data provider registrations, notification control, and provides interfaces for the enumeration of registered providers and devices that are providing information.		
127.	Workstation Service	It is a server-side user-mode process that provides an RPC-based network management interface to the Redirector.	wkssvc.dll (services.exe)	RPC

OS Support Component Decomposition

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
128.	DFS Service	It provides an LPC interface to the DFS Server Driver dfs.sys). Its purpose is to assist the DFS Server with name resolution.	dfssvc.exe	N/A
129.	DNS	DNS Servers (Name Severs) are server components that hold information about the domain tree's structure and configuration. DNS Clients (Resolvers) are programs that extract information from name servers in response to client requests.	dns.exe, dnsrslvr.dll, dnsapi.dll	RPC, network protocol
130.	Logical Disk Manager (LDM) Administrator service	It creates, interprets and modifies the contents of the LDM database.	dmadmin.exe	DCOM
131.	Network Connection Manager	It provides a controlling mechanism for all network connections managed by the host.	netman.dll (svchost.exe)	DCOM/RPC
132.	Print Spooler	It is responsible for spooling and printing of print jobs (documents) as well as maintaining all the system printer resources including printers, printer drivers, printer configuration data, the printer forms database, print processors, print jobs, and communicating with the network to allow remote printer administration and spooling of jobs remotely.	spoolss.exe, spoolss.dll, localspl.dll, win32spl.dll, tcpmon.dll, winprint.dll, unidrv.dll, brother.dll	RPC
133.	Session	It is responsible for the	smss.exe	LPC

	Manager	management of sessions and subsystems. A session is an environment that allows a user to log on and use the OS resources. A subsystem is an OS environment.		
134.	WMI Service	Windows Management Instrumentation (WMI) is an implementation of Web-Based Enterprise Management (WBEM), which is an industry initiative to develop a standard web-based technology for accessing management information. WMI uses the Common Information Model (CIM) industry standard to represent systems, applications, networks, devices, and other managed components.	winmgmt.exe, wbemcomn.dll, wbemcore.dll, fastprox.dll, wbemprox.dll, wbemsvc.dll, wbemdisp.dll, cimwin32.dll, dsprov.dll, ntevt.dll, provthrd.dll, secrcw32.dll, stdprov.dll, viewprov.dll, wbemess.dll, wbemperf.dll, wmiprov.dll	DCOM
135.	Windows Internet Naming System (WINS) service	It provides a distributed database for registering and querying dynamic NetBIOS names-to-IP address mappings in a routed network environment.	wins.exe, winsrpc.dll	RPC

RPC and Network Support Component Decomposition

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
136.	DCOM Services	It implements the Distributed Component Object Model (DCOM). DCOM extends the Component Object Model (COM) to support communication among objects on different computers over a local area network, a wide area network, or the Internet.	rpcss.dll (svchost.exe)	RPC
137.	RPC Endpoint Mapper	It allows RPC server applications to register their services as endpoints and RPC client applications to resolve binding information to locate endpoints for a given register service.	rpcss.dll (svchost.exe)	N/A

Administrator GUI Component Decomposition

	Subcomponent	Description	Binary Location	Type of external Interfaces implementing Security Checks
138.	ACL UI	Manages DACL and SACL on objects	aclui.dll	Graphical User Interfaces
139.	Active Directory Sites and Services	Manages AD Sites and Services	dsadmin.dll	Graphical User Interfaces
140.	Device Manager	Manages hardware devices installed on the local machine	devmgr.dll	Graphical User Interfaces
141.	Disk Admin	Manages disk and volumes	dmdskmgr.dll	Graphical User Interfaces
142.	Disk Quota	Manages disk quotas	dskquoui.dll	Graphical User Interfaces
143.	Domain and Trusts	Manages AD domains and trusts	domadmin.dll	Graphical User Interfaces
144.	Event Viewer	Views and manages security audit log	els.dll	Graphical User Interfaces
145.	Explorer	Manages files and directory	browseui.dll	Graphical User Interfaces
146.	Group Policy	Manages and configures group policy objects	localsec.dll, gpedit.dll	Graphical User Interfaces
147.	IPSEC Policy	Manages and configures IPSEC policy	ipsecsnp.dll	Graphical User Interfaces
148.	Network	Configures network protocols	netshell.dll	Graphical User Interfaces
149.	Network ID / Computer Name	Configures network ID or computer name	netid.dll	Graphical User Interfaces
150.	OU Delegation	Manages and configures OU delegation	dsuiwiz.dll	Graphical User Interfaces
151.	Registry Editor	Manages and configures registry keys	regedit.exe	Graphical User Interfaces
152.	Security Policy	Security policy administration	wsecedit.dll	Graphical User Interfaces
153.	Session Locking	Locking using screen saver	gina.dll	Graphical User Interfaces
154.	Time	Sets time	timedate.cpl	Graphical User Interfaces
155.	User, Groups, and Computers	Manages users and groups	dsadmin.dll	Graphical User Interfaces
156.	WMI Controls	Configures and controls WMI services	wbemcntl.dll	Graphical User Interfaces