SecureWave Sanctuary™ Application Control Desktop Security Target

ST Version 1.0

08 November 2006

Prepared For:

SecureWave
43869 Cowgill Court
Ashburn, VA 20147

Prepared By:

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

1 Security Target (ST) Introduction

Sanctuary™ Application Control Desktop is a three-tiered client/server application designed to allow or prevent execution of specific types of binaries depending on the binary file's contents.

This section 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, Vendor, and CC Identification

ST Title – SecureWave Sanctuary™ Application Control Desktop Security Target

ST Version – 1.0

TOE Identification – SecureWave Sanctuary™ Application Control Desktop version 2.8

Vendor – SecureWave

Evaluation Assurance Level (EAL) – EAL 2

Common Criteria Identification – Common Criteria for Information Technology Security Evaluation Version 2.1, August 1999, (International Standard – ISO/IEC 15408:1999)

1.2 Common Criteria Conformance Claims

This TOE and ST are consistent with the following specifications:

- Common Criteria (CC) for Information Technology Security Evaluation Part 2: Security functional requirements, Version 2.1, August 1999.
    - Part 2 Conformant
- Common Criteria (CC) for Information Technology Security Evaluation Part 3: Security assurance requirements, Version 2.1, August 1999,
    - Part 3 Conformant
    - Evaluation Assurance Level 2 (EAL2)

1.3 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, iteration is indicated by a letter in parenthesis placed at the end of the component. For example FCS_COP.1(a) and FCS_COP.1(b) indicate that the ST includes two iterations of the FCS_COP.1 requirement, a and b.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "…**big** ~~some~~ things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions and modifications made to comply with international interpretation.

1.4 Security Target Overview and Organization

This Sanctuary™ Application Control Desktop security target is organized as follows:

- Section 2 – Target of Evaluation (TOE) Description

  This section gives an overview of the TOE, describes the TOE in terms of its physical and logical boundaries, and states the scope of the TOE.

- Section 3 – TOE Security Environment

  This section details the expectations of the environment and the threats that are countered by Sanctuary™ Application Control Desktop and it's environment

- Section 4 – TOE Security Objectives

  This section details the security objectives of the Sanctuary™ Application Control Desktop and its environment.

- Section 5 – IT Security Requirements

  This section presents the security functional requirements (SFR) for Sanctuary™ Application Control Desktop and the environment that supports the TOE, and details the assurance requirements for EAL 2.

- Section 6 – TOE Summary Specification

  This section describes the security functions represented in the Sanctuary™ Application Control Desktop that satisfy the security requirements.

- Section 7 – Protection Profile Claims

  This section presents any protection profile claims.

- Section 8 – Rationale

  This section closes the ST with the justifications of the security objectives, requirements and TOE summary specifications as to their consistency, completeness and suitability.

2 Target of Evaluation (TOE) Description

The Target of Evaluation (TOE) is the SecureWave software application known as SecureWave Sanctuary™ Application Control Desktop, henceforth referred to as Sanctuary™ ACD. The TOE is a subset of the product. The product includes the MSDE 2000 database. The MSDE 2000 database is in the IT environment and not considered part of the TOE.

2.1 Product Type

Sanctuary™ ACD is a three-tiered client/server system designed to allow or prevent execution of specific types of executable files depending on the executable contents. The tiers are: a backend database (SQL Server); application server(s); and a client front end. The client front end comprises of the administrative clients, which is software used to control and direct the operation of the system, and the client drivers, residing on the computers that the Sanctuary™ ACD protects. The administrative client software resides in a main program and some smaller utility programs; the client drivers consist of one kernel driver each for NT 4.0, Windows 2000 and XP.

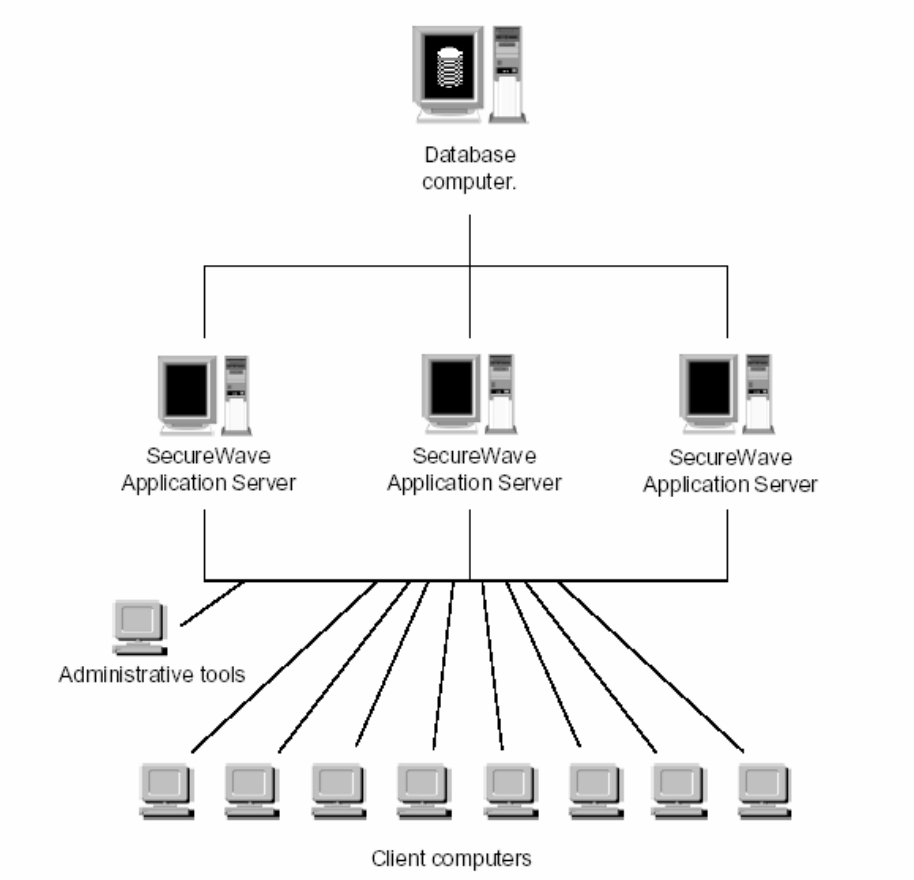

Figure 1: Client/Server architecture of the TOE.

2.2 Product Description

The fundamental rule used within the product is to allow only the use and/or execution of known and authorized executables and deny all else. In other words, the TOE does not use a “black list” of what is to

be prevented. It only uses a “white list” of what is allowed; everything else is denied by default. The product also authenticates, at every attempt to launch and/or use, that the “allowed” is valid.

The Sanctuary™ ACD uses a ‘white list’ of executable files that are allowed to run. The Sanctuary™ ACD also lets a system administrator decide which applications are allowed to run on the client computer systems. A SHA-1 hash of each authorized application and executable is recorded in a database from which a positive list is then derived. Every time a user decides to run an application, a local agent (kernel driver) will compare the application with the positive list. The positive list can be defined either in terms of a specific user or group of users. The Sanctuary™ ACD calculates a unique hash signature (based on SHA-1) for every binary executable file of the authorized applications. The Sanctuary™ ACD eliminates a wide range of threats and management problems including Trojan horses, viruses, games, suspicious downloads, and unlicensed software regardless of the source.

To achieve the desired protection, the server component of the TOE maintains a list of known and allowed executables, together with information on which user or user-group is allowed to run which executables. Also present in the database is information on the users and computers to be protected, as well as ancillary items.

Sanctuary™ ACD consists of four components which are described as follows:

- **Sanctuary™ Database -** This is the main storage point for the authorization information and is managed through the SecureWave Management Console. The database is hosted by Microsoft SQL Server 7/2000, MSDE or MSDE2000 and the underlying operating system. The TOE relies on the environment to provide Microsoft SQL Server 7/2000, MSDE or MSDE 2000 database for its use.
- **SecureWave Application Server -** SecureWave Application Server (SXS) communicates with the client computers and obtains from the Database the lists of files that the clients are permitted to run. SXS runs as a Windows service under any domain user account.
- **Sanctuary™ ACD Client Driver -** The Sanctuary™ ACD Client Driver (SXD) ensures that only the executable files that the user has been authorized to use can run on the computer. Any attempt to run an unauthorized file is barred and logged. The logs can be viewed using the SecureWave Management Console. The SXD provides interfaces that allow a user to authorize or deny the execution of a file and receive notification that access to a file has been denied. The SXD is installed on each client computer that will be controlled by the TOE.
- **Administrative Tools -** The Administration tools are utilized by the administrators to perform various administrative functions. The tools are SecureWave Management Console (SMC), Authorization Wizard, Key Pair Generator, and SXDomain command-line tool.

2.3 Product Features

The TOE implements the following features:

- **Protection against the “Unknown”** - Sanctuary™ ACD operates under the rule of “least privilege” principal and by default users have access to nothing. Administrators have to explicitly grant to each user/group the privilege to execute an application. Sanctuary™ ACD extends the operating system and immunizes it against executable threats. As a result organizations are able to exercise greater levels of security and control.
- **File integrity checking** - Sanctuary™ ACD works at a binary level. It examines each executable that an administrator wishes to authorize, and calculates a unique 20-byte digital signature (file signature) using a SHA-1 algorithm, based on the entire binary contents of that executable. The smallest change to the executable will result in a different file signature, which will mean that the altered file will not be able to run.
- **Software version control** - Sanctuary™ ACD recognizes files by their content rather than by their name or location. Thus the TOE is able to treat different versions of files as different files. The

result is that the administrator is able to control which file and the versions the users can run and allows the administrator full control over the process of upgrading the organization from one version of an application to another.

- **Prevents the installation of undesirable programs** - Sanctuary™ ACD also prevents the installation of undesirable programs as the installation program is an executable file and thus needs to be authorized to run.
- **Grant or revoke access on the fly** - The administrator may grant or revoke access to executables 'on the fly' for all logged in and connected users. There is no necessity for users to reboot or log off and then log on again for the changes to take effect. Users logged off and disconnected from the network will receive updates as soon as they reconnect and log into their system.
- **Logging of requests for executables** - Each time a client computer requests the use of an executable file, a log record is created. Sanctuary™ ACD provides the administrator the capability to review the details of the log records.
- **Protection at all times** - Sanctuary™ ACD affords organizations the same levels of control and protection regardless of whether the user or system is connected or isolated from the network. In standalone mode - when the client computer cannot communicate with the Sanctuary™ ACD Servers - it will consult its own database stored locally on the hard disk and consult the last list of authorized files. Sanctuary™ ACD will continue to use that list until it is able to connect to one of the Sanctuary™ ACD Servers to retrieve a later list.
- **Path Rules** - Sanctuary™ ACD includes the ability to manage applications based on their locations. Locations that are under the control of administrators, such as company network shares, are inherently safe. Path rules, when used in conjunction with hash rules, offer greater flexibility without compromising security. The Path Rules allow for control of the execution of a small number of applications for which hash checking will not work. The types of applications are executables that change as part of the installation procedure, or internal applications that change frequently and whose NTFS permissions are under the control of administrators.

2.4 Scope of TOE

2.4.1 Physical Boundary

Each component of the TOE is a software application that operates in a Microsoft Windows based environment. The physical boundary for each component of the TOE is the environment that each component requires for effective operation. This includes the database used for storage, the operating system and the hardware.

The TOE is able to operate effectively on the following platforms:

Application Servers (SXS)

- Operating Systems: Windows NT4 SP5 Server or Workstation, Windows 2000 Server or Professional, Windows Server 2003.
- Web Browsers: Internet Explorer 4.01 SP2 or later

Client, Admin Tools:

- Operating Systems: Windows NT4 SP6a Server or Workstation, Windows 2000 Server or Professional, Windows XP, Windows Server 2003.
- Web Browsers: Internet Explorer 5.0 or later

Database

- Operating Systems: Windows NT4 SP5 Server or Workstation, Windows 2000 Server or Professional, Windows XP, Windows Server 2003.

- Microsoft SQL Server (version 7.0 or above) or MSDE2000 (requires IE 5.0 or later)

The minimum hardware and other software requirements are identified in the assumption, A.HARDWRE.

2.4.2 Logical Boundary

The logical boundaries of the TOE can be described in terms of the security functions that the TOE implements to ensure that only authorized executable files are accessed.

2.4.2.1 Audit Function

The TOE records the actions that occur at the administrator and the client driver components. All administrative actions performed by the SMC are audited by the TOE. The SXD logs the actions of the client on the client computer. These logs are stored and protected by the operating environment of the client computer.

2.4.2.2 Cryptographic Function

The TOE utilizes a public-private key to sign the listings retrieved from the Database and sent to the client computers.

The TOE utilizes the SHA-1 Hash to create the digital signatures that are assigned to each executable file and that are created from the contents of the file. On the client computers, SHA-1 Hash is utilized to create digital signatures from the files the user attempts to execute. The resulting signatures are used for comparison against the authorized file signatures.

2.4.2.3 User Data Protection

Sanctuary™ ACD provides two methods for granting access to authorized executable files. One is based on matching the SXD-generated file signature to the authorized file signature assigned to an executable file.

The second method is the use of Path Rules that grant access to executable files and/or file directories based on a set of rules.

2.4.2.4 Security Management

The TOE provides the tool sets that are utilized by the administrator to manage and configure the security and administrative functions. These functions include the management of the executable files, the ability to manage the audit and log records, and the management of the access to the executable files.

The tool set consists of the following:

- **SecureWave Management Console (SMC)** - The SMC provides the administrative interface to SecureWave Application Server. It is used to configure Sanctuary™ ACD and carry out a range of day-to-day administrative tasks.
- **Authorization Wizard** - The Authorization Wizard can be used to identify the files that are present in the file directory, and to incorporate these files into the Database.
- **Key Pair Generator** - The Key Pair Generator is used to create an encryption key pair. The SXS uses an asymmetric encryption system to communicate with the SXD.
- **SXDomain command-line tool** - The SXDomain command-line tool provides an alternative method (other than using the SMC) for updating the Database with changes to the domains, users, groups and workstations within the network.

2.4.2.5 Protection of the TSF

The TOE implements security mechanisms to detect any tampering of the listing of file signatures and path rules that may have occurred during transmission of the listing from the SXS to the client's computer and the enforcement of the access control policy.

2.4.2.6 Resource Utilization

The TOE ensures that the access control policy is always enforced even if the client computer loses communication with SXS. The TOE stores the listing of the file signatures on the client computer, which is utilized to enforce the access control policy when a user attempts to access an executable file.

3 TOE Security Environment

The TOE security environment consists of the threats to the security of the TOE and usage assumptions as they relate to Sanctuary™ ACD. Sanctuary™ ACD provides for a level of protection that is appropriate for IT environments that require control over what applications and files are utilized by the users on the computer systems. It is suitable for use in both commercial and government environments.

3.1 Threats to Security

The following are threats identified for the TOE. The TOE itself has threats and the TOE is also responsible for addressing threats to the environment in which it resides.

T.ACCESS	An unauthorized user may gain access to the TOE and exploit system privileges to gain access to TOE security functions and data.
T.ACCOUNT	The security relevant actions of users may go undetected.
T.AUDIT_CORRUPT	Unauthorized users may tamper with audit data by gaining unauthorized access to the audit trail.
T.EXECUTE	A user may execute an unauthorized file or application.
T.PRIVILEGE	An unauthorized user may gain access to the TOE and alter the TOE configuration allowing for the execution of malicious applications and software.
T.TRANSIT	An unauthorized user may alter the TSF data as it is transmitted between the distributed parts of the TOE and the modification goes undetected.
T.FAULT	A user may attempt to execute unauthorized files or applications when communication between the client and server fails.

3.2 Secure Usage Assumptions

This section describes the security aspects of the environment in which the TOE will be utilized. This includes information about the physical, personnel, and system aspects of the environment.

3.2.1 Physical Assumptions

A.CONNECT	Any network resources used for communication between TOE components will be adequately protected from unauthorized access.
A.PROTECT	The database and server components must be located within controlled access facilities that will be protected from unauthorized physical access and modification.

3.2.2 Personnel Assumptions

A.MANAGE	There will be one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NOEVIL	The administrative personnel are not careless, willfully negligent, or hostile and will follow and abide by the instructions provided by the administrative guidance.

3.2.3 System Assumptions

A.HARDWRE The TOE components will be installed on a hardware system that meets or exceeds the following constraints:

	Application Server	Database	Admin Tools	Client

Operating System	Windows NT4 SP5 Server or Workstation, Windows 2000 Server or Professional, Windows Server 2003.	Windows NT4 SP5 Server or Workstation, Windows 2000 Server or Professional, Windows XP, Windows Server 2003.	Windows NT4 SP6a Server or Workstation, Windows 2000 Server or Professional, Windows XP, Windows Server 2003.	Windows NT4 SP4 Server or Workstation, Windows 2000 Server or Professional, Windows XP, Windows Server 2003.
Hard disk space	Program files: 1Mb Free disk space needed to install: 10Mb.	Program files: 5MB Free disk space needed to install: 40 MB Disk space for data: 20Mb+ (Depends on number of users)	Program files: 10Mb. Free disk space needed to install: 10Mb.	Program files: 5Mb Free disk space needed to install: 10Mb. Disk space for data: approx 10Mb
Memory	128Mb (256Mb recommended)	128Mb (256Mb recommended)	128Mb (256Mb recommended)	128Mb (256Mb recommended)
Deployment	Running setup.exe will install MSI 2.0 if not yet present. Using the MSI setup directly requires MSI 2.0 installed.			Using the MSI setup requires MSI 1.1.
Display Resolution	N/A	N/A	1024x768	N/A
File System	NTFS	NTFS	NTFS	NTFS
Other	MDAC V2.6 SP1 IE 4.01 SP2 or later. Setup will install MSI2.0 if not yet present.	Microsoft SQL Server (version 7.0 or above) or MSDE2000 (requires IE 5.0 or later MDAC V2.6 SP1 Setup will install MSI2.0 if not yet present.	Internet Explorer 5.0 or later Setup will install MSI2.0 if not yet present.	Using the MSI setup requires MSI 1.1, IE 4.01 SP2 or later.

A.IDENT The operating environment will provide a method of administrative identification and authentication.

A.SYSPRCT The operating environment will provide protection to the TOE and its related data.

A.SYSTIME The operating environment will provide reliable system time.

4 Security Objectives

This section describes the security objectives for the Sanctuary™ ACD and its supporting environment. Security objectives, categorized as either security objectives of the TOE or security objectives of the environment, reflect the stated intent to counter identified threats and assumptions. All of the identified threats and assumptions are addressed by the categories listed below.

4.1 Security Objectives of the TOE

The following security objectives are intended to be satisfied by the TOE.

O.CONTROL	The TSF must control access to executable files based on subject's identification. The TSF must provide the ability to limit each subject's access.
O.AUDIT	The TSF must record the security relevant actions of users of the TOE and have the ability to associate each action with a unique subject. The TSF must present this information in a readable format to authorized users and ensure that only authorized users are able to access this information.
O.DATA_TRANSFER	The TSF must have the capability to detect modifications to the TSF data transmitted between distributed parts of the TOE.
O.CRYPTO_KEYS	The TSF must ensure that cryptographic keys are generated in accordance with requirements defined by RSA.
O.CRYPTO_OPS	The TSF must ensure that all cryptographic operations used to protect information and generate file signatures meet the standards defined by RSA and FIPS 180 respectively.
O.FAULT_TOLERANCE	The TSF must continue to enforce access control policies if communications are lost with the central administration server.
O.MANAGE	The TOE must allow administrators to effectively manage the TOE and its security functions, and must ensure that only authorized administrators are able to access such functionality.

4.2 Security Objectives of the IT Environment

The following security objectives for the IT environment of the TOE must be satisfied in order for the TOE to fulfill its own security objectives.

OE.AUTH_ACCESS	The TOE operating environment must ensure that only authorized users gain access to the TOE and to the data contained in the TOE.
OE.ENV_ADMIN	The TOE operating environment must assign the administrative user to manage the TOE, until the TOE administrators are specifically assigned to manage the Administrative Tools component of the TOE.
OE.SEP	The TOE operating environment shall provide mechanisms to isolate the TOE Security Functions (TSF) and assure that TSF components cannot be tampered with or bypassed.
OE.ENV_FUNC	The TOE operating environment shall provide an accurate timestamp and protection of the stored TSF data.

4.3 Security Objectives of the Non - IT Environment

The following security objectives are intended to be satisfied by the environment of the TOE.

OE.CONNECT	Any network resources used for communication between TOE components will be adequately protected from unauthorized access.
OE.INSTALL	Those responsible for the TOE must ensure that the TOE is delivered, installed, managed, and operated in a manner that maintains the TOE security objectives.
OE.PERSON	Authorized users of the TOE shall be properly trained in the configuration and usage of the TOE and will follow the guidance provided.
OE.PHYCAL	Those responsible for the TOE must ensure that the parts of the TOE critical to security policy are protected from physical attack that might compromise the TOE security objectives.

5 IT Security Requirements

This section of the ST details the security functional requirements (SFRs) for the TOE and the IT Environment that will support the TOE.

5.1 TOE Security Functional Requirements

This section of the ST details the security functional requirements for the TOE. The SFRs are drawn from the CC Part 2. The following table lists the security functional requirements.

Security Functional Class	Security Functional Requirements
Security Audit (FAU)	FAU_GEN.1 Audit Data Generation
	FAU_GEN.2 User Identity Association
	FAU_SAR.1 Audit Review
	FAU_SAR.3 Selectable Audit Review
Cryptographic Support (FCS)	FCS_CKM.1 Cryptographic Key Generation
	FCS_COP.1 Cryptographic Operation
User Data Protection (FDP)	FDP_ACC.1 Subset Access Control
	FDP_ACF.1 Security attribute based access control
Identification and Authentication (FIA)	FIA_ATD.1 User Attribute Definition
Security management (FMT)	FMT_MOF.1 Management of security functions behaviour
	FMT_MSA.1 Management of Security Attributes
	FMT_MSA.3 Static attribute initialization
	FMT_MTD.1 Management of TSF data
	FMT_REV.1 Revocation
	FMT_SMF.1 Specification of Management Functions
	FMT_SMR.1 Security roles
Protection of the TSF (FPT)	FPT_ITT.3 TSF data integrity monitoring
	FPT_RVM.1 Non-bypassability of the TSP
Resource Utilization (FRU)	FRU_FLT.1 Degraded Fault Tolerance

Table 1: Security Functional Requirements

5.1.1 Security Audit (FAU)

5.1.1.1 FAU_GEN.1 Audit data generation

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

a) Start-up and shutdown of the audit functions;

b) All auditable events for the [***not specified***] level of audit***;*** and

c) [**see Table 2**].

Component	Event	Details
FDP_ACF.1	Enforcement of access control based upon security attributes	User, object
FMT_MOF.1	All modifications in the behavior of the TSF	
FMT_MSA.3	Modification of security attributes and their default values	
FMT_MTD.1	All modifications to the values of TSF data	User
FMT_REV.1	All attempts to revoke security attributes	

Table 2: Auditable Events

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

For each audit event type, based on the auditable event definitions of the functional components included in the ~~PP/~~ST, [**computer identity, filename**].

5.1.1.2 FAU_GEN.2 User identity association

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 FAU_SAR.1 Audit review

FAU_SAR.1.1 The TSF shall provide [**Enterprise Administrator, Administrator**] with the capability to read [**all audit information**] from the audit records.

FAU_SAR.1.2 The TSF shall provide the audit records in a manner suitable for the user to interpret the information.

5.1.1.4 FAU_SAR.3(a) Selectable audit review (Log)

FAU_SAR.3.1 The TSF shall provide the ability to perform [***searches***] of audit data **in the client log** based on [**date range, user identity, computer identity, filename**].

5.1.1.5 FAU_SAR.3(b) Selectable audit review (Audit)

FAU_SAR.3.1 The TSF shall provide the ability to perform [***searches***] of audit data **in the audit file** based on [**date range**].

5.1.1.6 FAU_SAR.3(c) Selectable audit review (Audit and log)

FAU_SAR.3.1 The TSF shall provide the ability to perform [***ordering***] of audit data based on [**date range, user identity, computer identity, filename**].

5.1.2 Cryptographic Support (FCS)

5.1.2.1 FCS_CKM.1 Cryptographic key generation (Data Encryption/Decryption)

FCS_CKM.1.1 The TSF shall generate cryptographic keys in accordance with a specified cryptographic key generation algorithm [**RSA key generation algorithm**] and specified cryptographic key sizes [**2048 bits asymmetric keys in BER format**] that meet the following: [**RSA: as defined by Rivest, Shamir, Adelman; BER as subset of ASN.1**.]

5.1.2.2 FCS_COP.1(a) Cryptographic operation (Data Encryption/Decryption)

FCS_COP.1.1 The TSF shall perform [**data encryption and decryption**] in accordance with a specified cryptographic algorithm [**RSA asymmetric crypto**] and cryptographic key sizes [**2048 bits asymmetric keys in BER format**] that meet the following: [**RSA: as defined by Rivest, Shamir, Adelman; BER as subset of ASN.1**.].

5.1.2.3 FCS_COP.1(b) Cryptographic operation (Hashing)

FCS_COP.1.1 The TSF shall perform [**hashing**] in accordance with a specified cryptographic algorithm [**SHA-1 Hash**] and cryptographic key sizes [**not applicable**] that meet the following: [**FIPS 180-1 Standard**].

5.1.3 User Data Protection (FDP)

5.1.3.1 FDP_ACC.1 Subset access control

FDP_ACC.1.1 The TSF shall enforce the [**Execution Control SFP**] on [**subject: user, object: executable file, operation: execution**].

5.1.3.2 FDP_ACF.1 Security attribute based access control

FDP_ACF.1.1 The TSF shall enforce the [**Execution Control SFP**] on objects based on **the following** [

Subject: user

- **user identity/user group**
- **file signatures,**
- **path rules**

Object: executable file

- **file signature**

][i]

FDP_ACF.1.2 The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed:

[**The user is able to execute the file:**

- **if the file signature of the executable file matches the user's generated**

[i] This requirement has been modified to comply with International Interpretation RI# 103.

file signature,

- **if the comparison fail, then execution is granted if the object's directory information matches the defined path rule.**]

FDP_ACF.1.3 The TSF shall explicitly authorize access of subjects to objects based on the following additional rules: [**if the user has been granted local authorization to explicitly authorize the execution of the file**].

FDP_ACF.1.4 The TSF shall explicitly deny access of subjects to objects based on the [**no additional explicit denial rules**]

5.1.4 Identification and Authentication (FIA)

5.1.4.1 FIA_ATD.1 User attribute definition

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

a) **user identity/user groups**,

b) **file signatures,**

c) **path rule**.]

5.1.5 Security Management (FMT)

5.1.5.1 FMT_MOF.1 Management of security functions behaviour (Access Control)

FMT_MOF.1.1 The TSF shall restrict the ability to [***enable, disable, modify the behaviour of***] the functions [**access control**] to [**Enterprise Administrator, Administrator**].

5.1.5.2 FMT_MSA.1(a) Management of security attributes (Files)

FMT_MSA.1.1 The TSF shall enforce the [**Execution Control SFP**] to restrict the ability to **[grant authorization**] the security attributes [**file signatures**] to [**Enterprise Administrator, Administrator**].

5.1.5.3 FMT_MSA.1(b) Management of security attributes (Path Rules)

FMT_MSA.1.1 The TSF shall enforce the [**Execution Control SFP**] to restrict the ability to **[*modify, delete,* create**, **add a user/user group, remove a user/user group**] the security attributes [**path rules**] to [**Enterprise Administrator, Administrator**].

5.1.5.4 FMT_MSA.3 Static attribute initialization

FMT_MSA.3.1 The TSF shall enforce the [**Execution Control SFP**] to provide [***restrictive***] default values for security attributes that are used to enforce the SFP.

FMT_MSA.3.2 The TSF shall allow the [**Enterprise Administrator, Administrator**] to specify alternative initial values to override the default values when an object or information is created.

5.1.5.5 FMT_MTD.1(a) Management of TSF data (Audit & Log Files)

FMT_MTD.1.1 The TSF shall restrict the ability to [***query***] the [**audit and log records**] to [**Enterprise Administrator, Administrator**].

5.1.5.6 FMT_MTD.1(b) Management of TSF data (Administrator)

FMT_MTD.1.1 The TSF shall restrict the ability to [**assign**] the [**Administrator**] to [**Enterprise Administrator**].

5.1.5.7 FMT_REV.1 Revocation

FMT_REV.1.1 The TSF shall restrict the ability to revoke security attributes associated with the [***subjects, objects***] within the TSC to [**Enterprise Administrator, Administrator**]

FMT_REV.1.2 The TSF shall enforce the rules [**as follows:**

- **If the server cannot communicate, immediately when the client connects to the network,**
- **If the server can communicate with the client:**
  - **Immediately, if the administrator pushes the updates to the client,**
  - **Otherwise the next time the user logs onto the client**].

5.1.5.8 FMT_SMF.1 Specification of Management Functions[ii]

FMT_SMF.1.1 The TSF shall be capable of performing the following security management functions: [

a) **grant authorization to executable files (file signatures),**

b) **modify, delete, create path rules,**

c) **review Audit and Logs Files,**

d) **ability to set local authorization.**].

5.1.5.9 FMT_SMR.1(a) Security roles

FMT_SMR.1.1 The TSF shall maintain the roles [**Enterprise Administrator, Administrator**].

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

5.1.6 Protection of the TSF (FPT)

5.1.6.1 FPT_ITT.3 TSF data integrity monitoring

FPT_ITT.3.1 The TSF shall be able to detect [***modification of data***] for TSF data transmitted between separate parts of the TOE.

FPT_ITT.3.2 Upon detection of a data integrity error, the TSF shall take the following actions: [**the client driver ignores the TSF data tranasmitted**.].

[ii] This requirement has been added to comply with International Interpretation #65

5.1.6.2 FPT_RVM.1 Non-bypassability of the TSP

FPT_RVM.1.1 The TSF shall ensure that TSP enforcement functions are invoked and succeed before each function within the TSC is allowed to proceed.

5.1.7 Resource Utilization (FRU)

5.1.7.1 FRU_FLT.1 Degraded Fault Tolerance

FRU_FLT.1.1 The TSF shall ensure the operation of [**Execution Control SFP**] when the following failures occur: [**the client driver is unable to communicate with the application server, the file signatures are tampered with**].

5.2 IT Environment Security Functional Requirements

This section details the security functional requirements of the IT Environment. The requirements drawn from the CC Part 2, address security functional dependencies the TOE has on the environment.

Security Functional Class	Security Functional Requirements
Security Audit (FAU)	FAU_STG.1 Protected audit trail storage
Identification and Authentication (FIA)	FIA_UID.2 User identification before any action
	FIA_UAU.2 User authentication before any action
Security management (FMT)	FMT_SMR.1 Security roles
Protection of the TSF (FPT)	FPT_SEP.1 TSF domain separation
	FPT_STM.1 Reliable time stamps

Table 3: Security Functional Requirements

5.2.1 Security Audit (FAU)

5.2.1.1 FAU_STG.1 Protected audit trail storage

FAU_STG.1.1 The **IT environment** shall protect the stored audit records from unauthorised deletion.

FAU_STG.1.2 The **IT environment** shall be able to [***prevent***] unauthorized modifications to the audit records in the audit trail.

5.2.2 Identification and Authentication (FIA)

5.2.2.1 FIA_UID.2 User identification before any action

FIA_UID.2.1 The **IT environment** shall require each user to identify itself before allowing any other TSF-mediated actions on the behalf of that user.

5.2.2.2 FIA_UAU.2 User authentication before any action

FIA_UAU.2.1 The **IT environment** shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on the behalf of that user.

5.2.3 Security Management (FMT)

5.2.3.1 FMT_SMR.1(b) Security roles

FMT_SMR.1.1 The **IT environment** shall maintain the roles [**Administrator**].

FMT_SMR.1.2 The **IT environment** shall be able to associate users with roles.

5.2.4 Protection of the TSF (FPT)

5.2.4.1 FPT_SEP.1 TSF domain separation

FPT_SEP.1.1 The **IT environment** shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

FPT_SEP.1.2 The **IT environment** shall enforce separation between the security domains of subjects in the TSC.

5.2.4.2 FPT_STM.1 Reliable time stamps

FPT_STM.1.1 The **IT environment** shall be able to provide reliable time stamps for its own use.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are the components included in Evaluation Assurance Level (EAL) 2 as specified in Part 3 of Common Criteria.

Assurance Class	Assurance Components
Class ACM: Configuration management	ACM_CAP.2 Configuration Items
Class ADO: Delivery and operation	ADO_DEL.1 Delivery Procedures
	ADO_IGS.1 Installation, generation, and start-up procedures
Class ADV: Development	ADV_FSP.1: Informal functional specification
	ADV_HLD.1: Descriptive high-level design
	ADV_RCR.1 Informal correspondence demonstration
Class AGD: Guidance documents	AGD_ADM.1 Administrator guidance
	AGD_USR.1 User guidance
Class ATE: Tests	ATE_COV.1: Evidence of coverage
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing - sample
Class AVA: Vulnerability assessment	AVA_SOF.1 Strength of TOE security function evaluation
	AVA_VLA.1 Developer vulnerability analysis

Table 4: Assurance Components for EAL 2

5.3.1 Class ACM: Configuration Management

5.3.1.1 ACM_CAP.2 - Configuration items

ACM_CAP.2.1D	The developer shall provide a reference for the TOE.
ACM_CAP.2.2D	The developer shall use a CM system.
ACM_CAP.2.3D	The developer shall provide CM documentation.
ACM_CAP.2.1C	The reference for the TOE shall be unique to each version of the TOE.
ACM_CAP.2.2C	The TOE shall be labeled with its reference.
ACM_CAP.2.3C	The CM documentation shall include a configuration list.
International Interpretation RI #3	**The configuration list shall uniquely identify all configuration items that comprise the TOE.**[iii]
ACM_CAP.2.4C	The configuration list shall describe the configuration items that comprise the TOE.
ACM_CAP.2.5C	The CM documentation shall describe the method used to uniquely identify the

[iii] This requirement has been added to comply with International Interpretation #3

configuration items.

ACM_CAP.2.6C The CM system shall uniquely identify all configuration items.

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

5.3.2 Class ADO: Delivery and Operation

5.3.2.1 ADO_DEL.1 - Delivery procedures

ADO_DEL.1.1D The developer shall document procedures for delivery of the TOE or parts of it to the user.

ADO_DEL.1.2D The developer shall use the delivery procedures.

ADO_DEL.1.1C The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.

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

5.3.2.2 ADO_IGS.1 Installation, generation, and start-up procedures

ADO_IGS.1.1D The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.

ADO_IGS.1.1C The **installation, generation and start-up** documentation shall describe **all** the steps necessary for secure installation, generation, and start-up of the TOE.[iv]

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

ADO_IGS.1.2E The evaluator shall determine that the installation, generation, and start-up procedures result in a secure configuration.

5.3.3 Class ADV: Development

5.3.3.1 ADV_FSP.1 - Informal functional specification

ADV_FSP.1.1D The developer shall provide a functional specification.

ADV_FSP.1.1C The functional specification shall describe the TSF and its external interfaces using an informal style.

ADV_FSP.1.2C The functional specification shall be internally consistent.

ADV_FSP.1.3C The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.

ADV_FSP.1.4C The functional specification shall completely represent the TSF.

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

ADV_FSP.1.2E The evaluator shall determine that the functional specification is an accurate and complete instantiation of the TOE security functional requirements.

[iv] This requirement has been modified to comply with International Interpretation #51.

5.3.3.2 ADV_HLD.1 - Descriptive high-level design

ADV_HLD.1.1D	The developer shall provide the high-level design of the TSF.
ADV_HLD.1.1C	The presentation of the high-level design shall be informal.
ADV_HLD.1.2C	The high-level design shall be internally consistent.
ADV_HLD.1.3C	The high-level design shall describe the structure of the TSF in terms of subsystems.
ADV_HLD.1.4C	The high-level design shall describe the security functionality provided by each subsystem of the TSF.
ADV_HLD.1.5C	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.
ADV_HLD.1.6C	The high-level design shall identify all interfaces to the subsystems of the TSF.
ADV_HLD.1.7C	The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.
ADV_HLD.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ADV_HLD.1.2E	The evaluator shall determine that the high-level design is an accurate and complete instantiation of the TOE security functional requirements.

5.3.3.3 ADV_RCR.1 Informal correspondence demonstration

ADV_RCR.1.1D	The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.
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.
ADV_RCR.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.4 Class AGD: Guidance Documents

5.3.4.1 AGD_ADM.1Administrator guidance

AGD_ADM.1.1D	The developer shall provide administrator guidance addressed to system administrative personnel.
AGD_ADM.1.1C	The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.
AGD_ADM.1.2C	The administrator guidance shall describe how to administer the TOE in a secure manner.
AGD_ADM.1.3C	The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.
AGD_ADM.1.4C	The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.
AGD_ADM.1.5C	The administrator guidance shall describe all security parameters under the control

of the administrator, indicating secure values as appropriate.

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.

AGD_ADM.1.7C The administrator guidance shall be consistent with all other documentation supplied for evaluation.

AGD_ADM.1.8C The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.

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

5.3.4.2 AGD_USR.1 User guidance

AGD_USR.1.1D The developer shall provide user guidance.

AGD_USR.1.1C The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.

AGD_USR.1.2C The user guidance shall describe the use of user-accessible security functions provided by the TOE.

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.

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 behaviour found in the statement of TOE security environment.

AGD_USR.1.5C The user guidance shall be consistent with all other documentation supplied for evaluation.

AGD_USR.1.6C The user guidance shall describe all security requirements for the IT environment that are relevant to the user.

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

5.3.5 Class ATE: Tests

5.3.5.1 ATE_COV.1 - Evidence of coverage

ATE_COV.1.1D The developer shall provide evidence of the test coverage.

ATE_COV.1.1C The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.

ATE_COV.1.1E The evaluator shall confirm that the information provided meets all requirements for content an-d presentation of evidence

5.3.5.2 ATE_FUN.1 Functional testing

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

ATE_FUN.1.2D The developer shall provide test documentation.

ATE_FUN.1.1C The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.

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.
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.
ATE_FUN.1.4C	The expected test results shall show the anticipated outputs from a successful execution of the tests.
ATE_FUN.1.5C	The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.
ATE_FUN.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.3.5.3 ATE_IND.2 Independent testing – sample

ATE_IND.2.1D	The developer shall provide the TOE for testing.
ATE_IND.2.1C	The TOE shall be suitable for testing.
ATE_IND.2.2C	The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.
ATE_IND.2.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
ATE_IND.2.2E	The evaluator shall test a subset of the TSF as appropriate to confirm that the TOE operates as specified.
ATE_IND.2.3E	The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.

5.3.6 Class AVA: Vulnerability Assessment

5.3.6.1 AVA_SOF.1 Strength of TOE security function evaluation

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.
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.
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.
AVA_SOF.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_SOF.1.2E	The evaluator shall confirm that the strength claims are correct.

5.3.6.2 AVA_VLA.1 - Developer vulnerability analysis[v]

AVA_VLA.1.1D	The developer shall perform **a vulnerability analysis**~~.~~

[v] This requirement has been modified to comply with International Interpretation #51

AVA_VLA.1.2D	The developer shall **provide vulnerability analysis documentation**.
AVA_VLA.1.1C	**The vulnerability analysis documentation shall describe the analysis of the TOE deliverables performed to search for obvious ways in which a user can violate the TSP.**
AVA_VLA.1.2C	**The vulnerability analysis documentation shall describe the disposition of obvious vulnerabilities.**
AVA_VLA.1.3C	The vulnerability analysis documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE
AVA_VLA.1.1E	The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.
AVA_VLA.1.2E	The evaluator shall conduct penetration testing, building on the developer vulnerability analysis, to ensure obvious vulnerabilities have been addressed.

6 TOE Summary Specification

6.1 TOE Security Functions

Each of the security function descriptions is organized by the security requirements corresponding to the security function. Hence, each function is described by describing how it specifically satisfies each of its related requirements. This serves to both describe the security functions and rationalize that the security functions are suitable to satisfy the necessary requirements.

6.1.1 Audit Function

FAU_GEN.1 Audit Data Generation

The TOE has an audit mechanism which generates audit records of the administrator's actions via the SMC. The following events are audited:

- Startup and shutdown of the client logging (audit) function: The ability to audit the Administrator's actions is always on, but the logging by the SXD is optional and can be enabled and disabled.
- All modification to the behavior of the TSF.
- All modifications of the default setting of restrictive values.
- All modifications to TSF data values, which includes the identification of the user who modified the data.
- All attempts to revoke the security attributes.

On each client computer the SXD logs the actions of the user when they attempt to execute a file (i.e, enforcement of the Execution Control SFP).

Each audit and log record contains the following information: date and time the event occurred, the type of event, identity of the user, the identity of the computer, the filename, and the success or failure of the event.

The client computers automatically transfer the logs to the SXS on a scheduled upload time or upon the request of the administrator.

FAU_GEN.2 User identity association

Each audit and log record generated is associated to the user, client or administrative, that performed the actions that triggered the generation of an audit or log record.

FAU_SAR.1 Audit Review

The TOE provides the ability for the administrator to view audit data and the log data for the system. The audit and log records are viewable via the SMC.

FAU_SAR.3 Selectable Audit Review

The SMC provides the ability for the administrator to search the audit records generated by the administrator's actions by a date range (FAU_SAR.3(b)). The log records generated by the users can be searched by date range, user identity, computer identity, and the file name (FAU_SAR.3(a)).

In addition, it is possible to order the records by clicking the column header with the required property of interest. The records can be ordered by the date, user identity, computer identity, file name, and other information available in the records. (FAU_SAR.3(c))

6.1.2 Cryptographic Function

FCS_CKM.1 Cryptographic key generation

The TOE uses the RSA key generation algorithm to generate a 2048-bit private-public key pair used for the encryption and decryption of the message digest. The private-public key pair is generated using the Keygen tool and stored in a BER format.

FCS_COP.1 Cryptographic operation

The SXS utilizes the private key to encrypt a message digest, which contains a hash of a user identity, a timestamp, and other ancillary information. The public key on the client computer is used to decrypt the message digest when it is received. The TOE performs the encryption and decryption using the RSA asymmetric crypto algorithm that meets the RSA standard.(FCS_COP.1(a))

The TOE utilizes the SHA-1 Hash to create the digital signatures that are assigned to each executable file and that are created from the contents of the file. On the client computers, SHA-1 Hash is utilized to create digital signatures from the files the user attempts to execute. The resulting signatures are used for comparison against the authorized file signatures. (FCS_COP.1(b))

6.1.3 User Data Protection

FDP_ACC.1 Subset access control, FIA_ATD.1 User Attribute Definition

The TOE creates a digital signature for each authorized executable file (file signature), and each executable file is associated to the users/user groups of the domain. The file signature, the file information, and the user's Security ID (SID) are stored in the Database for retrieval. In addition, the administrator is also able to assign path rules to a user or user group for executable files that change after installation or are internal applications that change constantly and whose NTFS permissions are under the control the administrator. These attributes are utilized by the Execution Control SFP to grant users access to executable files.

FDP_ACF.1 Security attribute based access control.

The Execution Control SFP by default enforces a deny access policy to any executable file until the administrator authorizes the executable file. The authorization of the file will cause the generation of the unique file signature to the file. The file signature is associated to all users/user groups in the domain and stored in the Database.

When a user logs onto a client computer for the first time, the listing of authorized file signatures associated with the user is retrieved from the Database and transmitted first to the SXS, and stored, then it is transmitted and stored on the client computer. When a user attempts to execute an executable file, the SXD generates a file signature using SHA-1 Hash and compares the signature to the authorized file signature provided from the Database. If there is a match, the file will execute.

In addition, the TOE offers an alternative function for controlling the execution of executable files. This function, known as Path Rules, allows for the user to execute the executable file if the file name and directory match the set of rules assigned by the administrator. The path rule can also be configured to verify that the user and/or user group is the owner of the file. The path rule will only allow execution if the user and/or user group is the owner of the file in addition to verifying the file. This path rule is only used when the file signature comparison fails or cannot be used because the files in question are frequently changed.

A user can be granted Local Authorization by the administrator, where the user will be able to explicitly authorize the execution of a file that is denied. This only grants local execution as the file signature is stored locally and not on the backend database.

6.1.4 Security Management

FMT_SMR.1 Security Roles

By default, the administrators of the TOE are users who are members of the local Administrators group defined by the OS on the servers running SXS. The SXS defines two administrator roles: Enterprise Administrator and Administrator and provides the ability to associate a user (defined in the environment) to the role of Enterprise Administrator and many users to the role of Administrator. The Enterprise Administrator and the Administrator are given the privilege to perform all administrative actions that are

provided by the SMC, with the exception that only an Enterprise Administrator is capable of assigning a user to the Administrator role.

FMT_MOF.1 Management of Security Functions Behavior

The SMC restricts the interfaces to configure and manage the TOE to the administrator defined in the local Administrator group of the IT environment or the users associated to the Enterprise Administrator and Administrator roles. The SMC provides the administrator with the interfaces to effect the behavior of the security and other administrative functions. The SMC also includes the following abilities:

- Authorize executable files that users can execute.
- The ability to grant user local authorization.
- Creation and modification of the Path Rules.
- Enable/Disable the Execution SFP for specific or all clients and for specific or all users/groups.

FMT_MSA.1 Management of security attributes

The SMC provides the administrator with the ability to determine which files can be executed, which causes the generation of a file signature, and create, modify, and delete the path rules. The file signatures and path rules ensure that the Execution Control SFP can be enforced based on the user identity and the file signature security attributes and the path rules that provide the user identity and the directory information used by the SFP.

FMT_MSA.3 Static attribute initialization

The SMC provides the administrator with the ability to determine what files can be executed. The TOE also provides the ability to alter the default values of the path rules. Initially, all execution is denied on the client computers until the administrator grants authorization to the executable files and a file signature is created.

FMT_MTD.1 Management of TSF Data

The SMC restricts the ability to review the audit and log records to the authorized administrator. The ability to assign a user to the Administrator role is restricted to the Enterprise Administrator.

FMT_REV.1 Revocation

The SMC provides the administrator with the ability to grant or revoke authorization of the files or create or modify the path rules. When updates are made, the administrator can send the updates to the specific users or all users that are logged in. When a user is not logged in or connected to the network at the time the updates are sent, they will receive the updates the next time they connect to the network and login. The updates will make changes to the cache as appropriate.

FMT_SMF.1 Specification of Management Functions

The SMC is divided into five modules, which provides the administrator with the graphical user interfaces that are used to configure and modify the options of the TOE. The modules are as follows:

- DB Explorer - utilized to show all the known executable files centrally authorized in the Database and allows the administrator the ability to delete files.
- EXE Explorer – utilized to build a list of executable files that can be centrally authorized.
- Log Explorer – displays the logs of all the actions that occurred on the client computers, providing the ability to search and order the logs.
- Scan Explorer – used to determine which executable files are present at different times on the same computer or on different computers, so that it can be determined which executables are required to run an application.
- Audit Logs Viewer – used to view the administrative audit records, providing the ability to search and order the records.

In addition to the modules, the SMC includes menu options that provide additional administrative functions.

6.1.5 Protection of TSF

FPT_ITT.3 TSF data integrity monitoring

The SXS creates a message digest, which includes a hash of the identity information about the client, timestamp and other information. The message digest is encrypted with the private key and then appended to the listing of the file signatures and the listing plus the message digest is sent to the client. The client utilizes the public key to decrypt the message digest, and then the client regenerates the hash from the data contained within the digest and makes a comparison. If the verification of the message digest by SXD fails, the listing of file signatures is ignored and the SXD requests the listing again from the SXS.

FPT_RVM.1 Non-bypassability of the TSP

The TOE implements the Execution Control SFP which by default denies the user ability execute an executable file until the administrator authorizes the executable files. The Execution Control SFP is always invoked when a user attempts to execute a file on a client machine that is protected by the TOE.

6.1.6 Resource Utilization

FRU_FLT.1 Degraded Fault Tolerance

The TOE enforces the Execution Control SFP regardless of whether the SXD (on the client's computer) can communicate with the SXS or if the file signatures are removed or corrupted.

SXS provides to client's computers the listings that determine what executable files the users have permission to execute. When the client's computer cannot communicate with the SXS, it will operate in a standalone mode, utilizing the copy of the listings placed on the hard disk of the computer. The SXD will utilize this listing until connection is reestablished and a new logon is performed or changes to the permissions were made.

If the listings are corrupted or removed from the client computer, the SXD denies access to all executable files.

6.2 Security Assurance Measures

The following assurance measures are applied to satisfy the Common Criteria EAL2 assurance requirements:

- Configuration Management;
- Delivery and Guidance;
- Development;
- Tests; and
- Vulnerability Assessment.

6.2.1 Configuration Management

The configuration management document describes the configuration management system utilized by SecureWave. The document describes the CM systems, identifies the configuration items that comprise the TOE and uniquely references each item and the TOE, and describes the procedure used to uniquely identify the TOE and the configuration items. SecureWave maintains configuration management control on the TOE implementation representation, design, tests, user and administrator guidance, and the CM documentation. These measures are described in:

- SecureWave IA Evaluation Program ACM_CAP.2 SecureWave Configuration Items

The configuration management documentation satisfies the ACM_CAP.2 assurance requirements.

6.2.2 Delivery and Guidance

6.2.2.1 Delivery and Installation

SecureWave provides delivery documentation that explains the procedures used to ensure that security is maintained when distributing the TOE to customers and installation and generation instructions at start-up. SecureWave's setup guide describes the steps to be used for the secure installation, generation, and start-up of the TOE along with configuration settings to secure the TOE privileges and functions. These procedures are documented in:

- ADO_DEL.1 SecureWave Delivery Process
- Sanctuary Application Control Desktop Setup Guide v2.8, May 14, 2004

The delivery and installation documentation satisfies the following assurance requirements:

- ADO_DEL.1,
- ADO_IGS.1.

6.2.2.2 Administrative and User Guidance

SecureWave provides administrator guidance on how to utilize the TOE security functions, other administrative functions and warnings to authorized administrators about actions that can compromise the security of the TOE. The procedures, included in the administrator guidance, describe the steps necessary to operate Sanctuary™ ACD in accordance with the evaluated configuration, detailing how to establish and maintain the secure configuration.

The administrator guidance is documented in:

- Sanctuary™ Application Control Desktop Administrator's Guide v2.8, May 13, 2004

The user guidance is documented in:

- Online Help

The administrator's guide and the online help satisfy the following assurance requirements:

- AGD_ADM.1,
- AGD_USR.1.

6.2.3 Development

The design documentation provides for Sanctuary™ ACD is provided in two documents:

- SecureWave EAL2 Functional Specification Sanctuary Application Control Desktop and Custom Edition
- SecureWave High Level Design,

These documents serve to describe the security functions of the TOE, its interfaces both external and between subsystems, the architecture of the TOE (in terms of subsystems), and correspondence between the available design abstractions (including the ST).

The design documentation satisfies the following security assurance requirements:

- ADV_FSP.1;
- ADV_HLD.1; and,
- ADV_RCR.1.

6.2.4 Tests

The test documentation is found in the following documents:

- SecureWave QA Test Plan Structure & Strategy
- SecureWave QA Test Environment Setup
- SACCE Test Plan
- Mapping

These documents describe the overall test plan, testing procedures, and the tests themselves, including expected and actual results. In addition, these documents describe how the functional specification has been appropriately tested.

The test documentation satisfies the following assurance requirements:

- ATE_COV.1;
- ATE_FUN.1; and,
- ATE_IND.2.

6.2.5 Vulnerability Assessment

SecureWave performed a vulnerability analysis of the TOE to identify weaknesses that can be exploited in the TOE. The vulnerability analysis is documented in:

- SecureWave Vulnerability Analysis

The vulnerability analysis documentation satisfies the following assurance requirements:

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

7 Protection Profile Claims

This TOE does not claim conformance to a Protection Profile.

8 Rationale

This section provides the rationale for completeness and consistency of this Security Target. The rationale addresses the following areas:

- Security Objectives;
- Security Requirements;
- Security Functional Requirement Dependencies;
- TOE Summary Specification;
- Strength of Function; and
- Internal Consistency.

8.1 Security Objectives Rationale

This section demonstrates that all threats and secure usage assumptions are completely covered by security objectives. In addition, each objective counters or addresses at least one assumption and threat.

Objectives	O.CONTROL	O.AUDIT	O.DATA_TRANSFER	O.CRYPTO_KEYS	O.CRYPTO_OPS	O.FAULT_TOLERANCE	O.MANAGE	OE.AUTH_ACCESS	OE.ENV_ADMIN	OE.SEP	OE.ENV_FUNC	OE.CONNECT	OE.INSTALL	OE.PERSON	OE.PHYCAL
A.CONNECT												X			X
A.HARDWRE													X		
A.IDENT								X							
A.MANAGE								X	X					X	
A.NOEVIL														X	
A.PROTECT															X
A.SYSPRCT								X		X	X				X
A.SYSTIME											X				
T.ACCOUNT		X					X	X							
T.EXECUTE	X				X										
T.FAULT						X									
T.TRANSIT			X	X	X										X
T.ACCESS		X					X	X							X
T.PRIVILEGE	X	X					X	X		X					X
T.AUDIT_CORRUPT								X		X					X

Table 5: Threats and Assumptions vs. Security Objectives

8.1.1 Security Objectives for the TOE

The following describe how the Security Objectives for the TOE and the Environment completely and effectively counters the threats that are implemented:

O.CONTROL	This objective ensures that the TSF controls access to the executables file based on the user's identity thus mitigating T.EXECUTE, and assists with mitigating T.PRIVILEGE.
O.AUDIT	This objective ensures that the TOE monitors and audits the actions of the administrators and the logs the actions of the users, and the provides the tools necessary to configure and manage the auditing and logging functions, thus assists in countering T.ACCESS, T.ACCOUNT, and T.PRIVILEGE.
O.DATA_TRANSFER	This objective ensures that TOE detects modifications made the file signatures transmitted between the SXS and the client computer, thus assists in mitigating T.TRANSIT.
O.CRYPTO_KEYS	This objective assists with the mitigation of T.TRANSIT by ensuring that the TOE generates the cryptographic key used to decrypt and encrypt the digest attached to the data transmitted between the SXS and the client computer.
O.CRYPTO_OPS	This objective assists with the mitigation of T.TRANSIT and T.EXECUTE by ensuring that the TOE performs the cryptographic operation of hashing to create and verify the file signatures used to determine if a user can execute a file or application and cryptographic operation of encryption and decryption of the digest attached to the data transmitted between the SXS and the client computer.
O.FAULT_TOLERANCE	This objective supports the mitigation of T.FAULT as it ensures that the access control function is still enforce in the event of the lost of connection to the SXS.
O.MANAGE	This objective ensures that TOE provides the functions and tools necessary to support the authorized administrator in managing TOE security are provided, assists in countering T.ACCESS, T.ACCOUNT, and T.PRIVILEGE because it requires the TOE to provide functionality to support the management of audit, access protection and other administrative functions.

8.1.2 Security Objectives for the Environment

This section provides evidence demonstrating the coverage of threats and assumptions by the environment security objectives.

8.1.2.1 Security Objectives for the IT Environment

This section demonstrates how the IT environment security objectives are effect in addressing the assumptions on the environment and the how they assist in the mitigation of threats against the TOE.

OE.AUTH_ACCESS	This objective ensures that the IT environment provides the identification and authentication mechanism that which ensures that only authorized users have access to the TOE and its associated data. This objective assists with mitigating T.ACCESS, T.ACCOUNT, T.PRIVILEGE and T.AUDIT_CORRUPT and addresses A.IDENT, A.MANAGE, and A.SYSPRCT.
OE.ENV_ADMIN	This objective requires that the IT environment define competent and

	trained users as administrators that can be utilize by the TOE, thus addressing A.MANAGE.
OE.SEP	This objective provides the support needed by the TOE to counter threats T.PRIVILEGE, and T.AUDIT_CORRUPT and addresses A.SYSPRCT by ensuring that the TOE and its associated data cannot be tampered with or bypassed.
OE.ENV_FUNC	This objective ensures that an accurate timestamp is provided; accurate record information on a time/date basis can be generated, queried and tracked and it ensures that the file signatures and the log files are protected. This objective addresses A.SYSTIME and A.SYSPRCT.

8.1.2.2 Security Objectives for the Non-IT Environment

This section demonstrates how the non-IT environment security objectives are effect in addressing the assumptions on the environment and the how they assist in the mitigation of threats against the TOE.

OE.CONNECT	This objective ensures that the network connection between the TOE components is protected. This objective addresses A.CONNECT.
OE.INSTALL	By ensuring that the TOE is delivered, installed, managed, and operated in a secure manner, the assumption A.HARDWRE is addressed. This objective ensures that the TOE is installed, configured, managed and administered in a secure manner by a competent and security aware individual in accordance with the administrator, delivery and installation documentation.
OE.PERSON	This objective ensures that the TOE is managed and administered in a secure manner by competent and security aware personnel in accordance with the administrator documentation, addressing A.NOEVIL, and A.MANAGE.
OE.PHYCAL	This objective ensures that the TOE is operated in an environment that will protect it from unauthorized access and physical threats and attacks that can disturb and corrupt the information generated. This objective addresses A.CONNECT, A.PROTECT, A.SYSPRCT and assists in countering T.ACCESS, T.TRANSIT, T.PRIVILEGE, and T.AUDIT_CORRUPT.
OE.CONNECT	This objective ensures that the network connection between the TOE components is protected. This objective addresses A.CONNECT.
OE.INSTALL	By ensuring that the TOE is delivered, installed, managed, and operated in a secure manner, the assumption A.HARDWRE is addressed. This objective ensures that the TOE is installed, configured, managed and administered in a secure manner by a competent and security aware individual in accordance with the administrator, delivery and installation documentation.
OE.PERSON	This objective ensures that the TOE is managed and administered in a secure manner by competent and security aware personnel in accordance with the administrator documentation, addressing A.NOEVIL, and A.MANAGE.
OE.PHYCAL	This objective ensures that the TOE is operated in an environment that will protect it from unauthorized access and physical threats and attacks that can disturb and corrupt the information generated. This objective addresses A.CONNECT, A.PROTECT, A.SYSPRCT and assists in countering T.ACCESS, T.TRANSIT, T.PRIVILEGE, and T.AUDIT_CORRUPT.

8.2 Security Requirements Rationale

This section provides evidence supporting the internal consistency and completeness of the components (requirements) in the Security Target. Note that Table 6: Security Functional Requirements vs. Security Objectives indicates the requirements that effectively satisfy the individual objectives. Objectives for the IT environment are satisfied only by requirements for the IT environment. However, some of those requirements also support, in some relatively small way, the TOE security objectives.

8.2.1 Security Functional Requirements Rationale

All Security Functional Requirements (SFR) identified in this Security Target are fully addressed in this section and each SFR is mapped to the objective it satisfies.

Objectives / Functional Requirements	O.CONTROL	O.AUDIT	O.DATA_TRANSFER	O.CRYPTO_KEYS	O.CRYPTO_OPS	O.FAULT_TOLERANCE	O.MANAGE	OE.AUTH_ACCESS	OE.ENV_ADMIN	OE.SEP	OE.ENV_FUNC
FAU_GEN.1		X									
FAU_GEN.2		X									
FAU_SAR.1		X									
FAU_SAR.3(a)		X									
FAU_SAR.3(b)		X									
FAU_SAR.3(c)		X									
FAU_STG.1											X
FCS_CKM.1			X	X	X						
FCS_COP.1(a)			X		X						
FCS_COP.1(b)			X		X						
FDP_ACC.1	X										
FDP_ACF.1	X										
FIA_ATD.1	X										
FIA_UID.2								X	X		
FIA_UAU.2								X			
FMT_MOF.1							X				
FMT_MSA.1(a)							X				
FMT_MSA.1(b)							X				
FMT_MSA.3							X				
FMT_MTD.1(a)							X				
FMT_MTD.1(b)							X				
FMT_REV.1							X				
FMT_SMF.1		X					X				
FMT_SMR.1(a)							X				
FMT_SMR.1(b)									X		
FPT_ITT.3			X								
FPT_RVM.1	X										
FPT_SEP.1										X	
FPT_STM.1											X
FRU_FLT.1						X					

Table 6: Security Functional Requirements vs. Security Objectives

The following text describes how each security objective is satisfied by the SFRs:

O.CONTROL	*The TSF must control access to executable files based on subject's identification. The TSF must provide the ability to limit each subject's access.*

The TOE is required to enforce the Execution Control SFP, which grants user's access to executable files by matching the generated file signatures to the authorized file signatures that are associated to the user, or if the executable files meet the criteria of the path rules assigned to the user. (FDP_ACC.1, FDP_ACF.1). The attributes required by the Execution Control SFP are stored in the Database (FIA_ATD.1), and are retrieved and cached by the SXS and the client computer. The TOE must ensure that the user is not granted access to an executable file until the file is authorized by the administrator (FPT_RVM.1).

O.AUDIT	*The TSF must record the security relevant actions of users of the TOE and have the ability to associate each action with a unique subject. The TSF must present this information in a readable format to authorized users and ensure that only authorized users are able to access this information*

The TOE audits all actions of the administrator and logs the actions of the users on the client computers, generating audit and log records (FAU_GEN.1), which are associated with the user that performed the actions (FAU_GEN.2). The TOE also provides a set of tools that are accessible to the administrator to review the audit and log data (FAU_SAR.1, FAU_SAR.3(a), FAU_SAR.3(b), FAU_SAR.3(c), FMT_SMF.1).

O.DATA_TRANSFER	*The TSF must have the capability to detect modifications to the TSF data transmitted between distributed parts of the TOE*

The TOE ensures any modifications made to the listing of authorized file signatures during transmission between the TOE components are detected by attaching an encrypted message digest to the listing (FPT_ITT.3), which is decrypted after the transmission and the enclosed hash is matched to the generated hash (FCS_COP.1(a), FCS_COP.1(b)). The encryption, decryption, and hashing utilize the key generated in accordance with RSA (FCS_CKM.1).

O.CRYPTO_KEYS	*The TSF must ensure that cryptographic keys are generated in accordance with requirements defined by RSA.*

The TOE generates the public-private key pair utilized to encrypt and decrypt the message digest, and the keys used in the hashing process in accordance with RSA (FCS_CKM.1).

O.CRYPTO_OPS	*The TSF must ensure that all cryptographic operations used to protect information and generate file signatures meet the standards defined by RSA and FIPS 180 respectively.*

The TOE generates the file signatures, and data of the message digest by the SHA-1 Hash method, which is performed in accordance with FIPS 180 (FCS_COP.1(b)), and encrypts and decrypts the message digest attached to the data transmitted from the SXS to the client computer in accordance with RSA (FCS_COP.1(a)). The TOE utilizes the keys generated in accordance with RSA to perform the encryption and decryption. (FCS_CKM.1)

O.FAULT_TOLERANCE	*The TSF must continue to enforce access control policies if communications are lost with the central administration server.*

The TOE ensures that the access control of the executable files is still enforced when the communications with the SXS is lost by caching the listing of authorized file signatures on the SXS and the client computers (FRU_FLT.1).

O.MANAGE	*The TOE must allow administrators to effectively manage the TOE and its security functions, and must ensure that only authorized administrators are able to access such functionality.*

The TOE is required to restrict the management of the access control function, which includes modification of the behaviour of the function (FMT_MOF.1), the ability to manage the security attributes of the SFP (FMT_MSA.1(a), FMT_MSA.1(b), FMT_MSA.3), the management of the audit and log records (FMT_MTD.1(a)), the ability to revoke the access granted to users at a moments notice (FMT_REV.1) to the administrators of the TOE. The Administrative Tools provides a set of tools that allow the administrators to configure and manage the TOE (FMT_SMF.1). The TOE must provide the ability to identify the administrators who can manage the TOE (FMT_SMR.1(a)). The Enterprise Administrator has the additional ability to assign users to the Administrator role (FMT_MTD.1(b)) Each of the mentioned requirements together ensures that the administrators are able to manage the TOE.

OE.AUTH_ACCESS	*The TOE operating environment must ensure that only authorized users gain access to the TOE and to the data contained in the TOE.*

Requiring that the TOE operating environment provide the identification and authentication mechanism to ensure that only authorized administrators have access to the TOE and its associated data satisfies this objective. (FIA_UID.2, FIA_UAU.2)

OE.ENV_ADMIN	*The TOE operating environment must provide the administrator to manage the TOE, until the TOE administrators are specially assigned to manage the Administrative Tools component of the TOE.*

Requiring that TOE operating environment defines the administrator role, thus providing the authorized administrators who will have access to the TOE and its associated data, satisfies this objective. (FMT_SMR.1(b))

OE.SEP	*The TOE operating environment shall provide mechanisms to isolate the TOE Security Functions (TSF) and assure that TSF components cannot be tampered with or bypassed.*

Requiring that the IT Environment protect the TOE from untrusted process that could attempt to tamper with or bypass the TOE satisfies this objective. (FPT_SEP.1)

OE.ENV_FUNC	*The TOE operating environment shall provide an accurate timestamp and protection of the stored TSF data.*

Requiring that the IT Environment provides accurate and reliable time mechanism (FPT_STM.1) and provide the protection of the log data stored on the client's computer (FAU_STG.1), satisfies this objective.

8.2.2 Security Functional Requirement Dependency Rationale

The dependencies of the TOE security functional requirements are met through the functionality of the TOE and/or by the security functionality of the IT environment.

The table below maps the TOE security functional requirements to the corresponding requirements they are dependent on. The Table demonstrates that all TOE security functional requirement dependencies are met within the ST. A rationale has been provided that addresses the requirements that were not addressed in the Table 7. Note: the table below assumes the requirement iterations have the same dependencies and therefore the iterations are not individually identified in the table (e.g. FCS_COP.1(a)).

Dependency / Functional Requirements	FAU_GEN.1	FAU_SAR.1	FCS_CKM.1	FCS_COP.1	FDP_ACC.1	FDP_ACF.1	FMT_MSA.1	FMT_MSA.3	FMT_SMF.1	FMT_SMR.1	FIA_UID.2	FPT_STM.1
FAU_GEN.1												X
FAU_GEN.2	X										X	
FAU_SAR.1	X											
FAU_SAR.3		X										
FCS_CKM.1				X								
FCS_COP.1			X									
FDP_ACC.1						X						
FDP_ACF.1					X			X				
FMT_MOF.									X	X		
FMT_MSA.1					X				X	X		
FMT_MSA.3							X			X		
FMT_MTD.1									X	X		
FMT_REV.1										X		
FMT_SMR.1											X	

Table 7: Security Functional Requirements Dependencies

For the FCS_COP.1 and FCS_CKM.1 requirements, the CC identifies the following dependencies; FCS_CKM.4, and FMT_MSA.2. The dependencies for the requirements are not applicable and the rationale is as follows:

- FCS_CKM.4: this requirement is concerned with the destruction of cryptographic keys. This requirement is not applicable to the TOE because the TOE uses a single key pair. For this TOE there is only one key pair that is generated used the system that the TOE monitors. This key pair is utilized for the encryption and decryption of the message data. This key pair is only generated once during the installation, and stored in BER Format and is utilized until the TOE is uninstalled from the system, where the process deletes that associated key file. The hashing operation does not utilize a cryptographic key, thus requirement is not applicable.
- FMT_MSA.2: this requirement is concerned with ensuring that only secure values are accepted for security attributes. This requirement is not applicable because the hashing does not utilize a cryptographic key and the key pair used for data encryption is generated by an administrative tool; Key Pair Generator, which is only accessible during installation. During the operation of the TOE, secure security attributes are not required as the TOE does not re-generate the key pair.

For FCS_COP.1(b), the FCS_CKM.1 dependency was not included in the ST, because the hash function does not utilize a cryptographic key, thus this requirement is not applicable.

For FPT_ITT.3, the FPT_ITT.1 dependency was not included in the ST, because this requirement is concerned with ensuring that the TSF protects the data during transmission, and the TOE rather than protect against the modification during transmission, always performs a verification of the transmitted TSF data to detect any modifications that may have occurred. Thus for this TOE this requirement is not applicable.

For FRU_FLS.1, the FPT_FLS.1 dependency was not included in the ST, because the FRU_FLS.1 requirement identifies failure conditions that are outside the scope f the TSF. The loss of communication between the client and SXS and the tampering of the file signatures are not failure conditions of the TSF and as such FPT_FLS.1 is not applicable to the TOE.

8.2.3 Security Assurance Requirements Rationale

This ST contains the assurance requirements from the CC EAL2 assurance package. The EAL chosen is based on the statement of the security environment (assumptions, threats and organizational policy) and the security objectives defined in this ST. The sufficiency of the EAL chosen (EAL2) is justified based on those aspects of the environment that impact upon the assurance needed in the TOE. The administrative staff is conscientious, non-hostile and well trained (A.NOEVIL, A.MANAGE, OE.PERSON). The TOE is physically protected (OE.PHYCAL), and properly and securely configured (OE.INSTALL). Given these aspects, a TOE based on good commercial development practices is sufficient. EAL 2 is an appropriate level of assurance for the TOE described in this ST.

8.3 TOE Summary Specification Rationale

This section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions and security assurance measures are suitable to meet the TOE security requirements. The collection of security functions works together to provide all of the security requirements. The security functions described in the TOE summary specification and indicated in Table 8: Security Functional Requirements vs. Security Functions are all necessary for the required functionalities in the TSF.

Table 9: Security Assurance Requirements vs. Assurance Measures provides a mapping of TOE security assurance functions to those security assurance measures that have been implemented by the developer to ensure that the TOE meets the requirements specified by CC EAL2.

Security Functions / Functional Requirements	AUDIT	CRYPTOGRAPHIC	USER DATA PROTECTION	SECURITY MANAGMENT	TSF PROTECTION	RESOURCE UTILIZATION
FAU_GEN.1	X					
FAU_GEN.2	X					
FAU_SAR.1	X					
FAU_SAR.3(a)	X					
FAU_SAR.3(b)	X					
FAU_SAR.3(c)	X					
FCS_CKM.1		X				
FCS_COP.1(a)		X				
FCS_COP.1(b)		X				
FDP_ACC.1			X			
FDP_ACF.1			X			
FIA_ATD.1			X			
FMT_MOF.1				X		
FMT_MSA.1(a)				X		
FMT_MSA.1(b)				X		
FMT_MSA.3				X		
FMT_MTD.1(a)				X		
FMT_MTD.1(b)				X		

Security Functions / Functional Requirements	AUDIT	CRYPTOGRAPHIC	USER DATA PROTECTION	SECURITY MANAGMENT	TSF PROTECTION	RESOURCE UTILIZATION
FMT_REV.1				X		
FMT_SMF.1				X		
FMT_SMR.1				X		
FPT_ITT.1					X	
FPT_RVM.1					X	
FRU_FLT.1						X

Table 8: Security Functional Requirements vs. Security Functions

Assurance Measures / Assurance Requirements	CONFIGURATION MANAGEMENT	DELIVERY AND GUIDANCE	DEVELOPMENT	TESTS	VULNERABILITY ASSESSMENT
ACM_CAP.2	X				
ADO_DEL.1		X			
ADO_IGS.1		X			
ADV_FSP.1			X		
ADV_HLD.1			X		
ADV_RCR.1			X		
AGD_ADM.1		X			
AGD_USR.1		X			
ATE_COV.1				X	
ATE_FUN.1				X	
ATE_IND.2				X	
AVA_SOF.1					X
AVA_VLA.1					X

Table 9: Security Assurance Requirements vs. Assurance Measures

8.4 Strength of Function Rationale

The only permutational and probabilistic mechanisms implemented in the TOE are the cryptographic mechanism defined by FCS_CKM.1 and FCS_COP.1. These mechanisms are outside the scope of the

evaluation. The TOE does not identify any other functions that are of a permutational or probabilistic nature. Therefore, a minimum SOF claim is not included for the TOE.

8.5 Internal Consistency and Support

The selected functional requirements for the TOE and IT Environment are internally consistent. All the operations performed are in accordance with the CC. The ST does not include any instances of a requirement that conflicts with or contradicts another requirement. In instances where multiple requirements apply to the same functions, the requirements and their operations do not cause a conflict between each other.

The selected requirements are mutually supportive by supporting the dependencies, the rationale of the suitability of the requirements to meet the objectives; the inclusion of architectural requirements, FPT_RVM.1 and FPT_SEP.1, to protect the TOE; the inclusion of audit requirements to detect attacks and the inclusion of management requirements to provide a means to properly configure and manage the other security requirements

Notes on Deviations

Note 1: In FMT_MSA.3 and FRU_FLT, corrected “initialisation” for spelling error.

Acronyms

ACD	Application Control Desktop
CC	Common Criteria
EAL	Evaluation Assurance Level
IT	Information Technology
MDAC	Microsoft Data Access Components.
MSDE	Microsoft SQL Server Desktop Engine
MSI	Microsoft Installer
SID	Windows Security ID
SFP	Security Function Policy
SFR	Security Function Requirement
SMC	SecureWave Management Console
ST	Security Target
SXD	SecureWave Client Driver
SXS	SecureWave Application Server
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions
TSP	TOE Security Policy

Terminology

CAB	File extension for cabinet files, which are multiple files compressed into one and extractable with the extract.exe utility. Such files are frequently found on Microsoft software distribution disks.
Client Computer	The computers on your network that Sanctuary™ ACD controls.
Dependencies	Additional executable files (.exe's or .dll's) that are required by executable files to run properly. Dependencies are split into two categories: static dependencies which are files declared explicitly in the executable file as being required, and dynamic dependencies which are additional files an executable may require at runtime.
Executable Program	A program that can be run. The term usually applies to a compiled program translated into computer code in a format that can be loaded in memory and run by a computer's processor.
Hash	A complex digital signature calculated by Sanctuary™ ACD to uniquely identify each executable file that can be run. The hash is calculated using the SHA-1 algorithm, which takes into account the entire contents of the file.
MSDE	A SQL Server 7-compatible database server, suitable for small and medium size organizations. MSDE is supplied with some versions of Sanctuary™ ACD. MSDE databases can subsequently be migrated to SQL Server 7.
MSI	Microsoft Installer package utilized to install the components for the TOE.
Private Key	One of two keys used in public key encryption. The sender uses the private key to create a unique electronic number that can be read by anyone possessing the corresponding public key, which verifies that the message is truly from the sender.
Public Key	One of two keys in public key encryption. The user releases this key to the public, who can use it for encrypting messages to be sent to the user and for decrypting the user's digital signature.
SHA-1	Secure Hash Algorithm 1, as defined in the Federal Information Processing Standards Publication 180-1. This algorithm produces a one-way 160-bit hash that can be used for a variety of applications including authentication and cryptography.
SID	MS Windows user identification
Security Target (ST)	A set of security requirements and specifications to be used as the basis for evaluation of an identified TOE.
Target of Evaluation (TOE)	An IT product of system and its associated guidance documentation that is the subject of an evaluation.
TCP/IP	The protocol used by the client computers to communicate with the SecureWave Application Server.