Cisco Security Agent Version 4.5.1 Security Target
April 4, 2007
Document No. EDCS-507896

Prepared By:

COACT, Inc.
9140 Guilford Road, Suite N
Columbia, Maryland 21046-2587

Prepared For:

Cisco Systems
1414 Massachusetts Ave.
Boxborough, MA 01719

DOCUMENT INTRODUCTION

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), the Cisco Security Agent Version 4.5.1. This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements and the IT security functions provided by the TOE which meet the set of requirements.

REVISION HISTORY

Rev	Description
1	April 4, 2007 Initial release

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMNS

CC
Common Criteria2

CSA
Cisco Security Agent1

EAL
Evaluation Assurance Level1

I&A
Identification & Authentication3

IT
Information Technology1

MC
Management Center3

NIAP
National Information Assurance Partnership1

PP
Protection Profile2

ST
Security Target1

TOE
Target of Evaluation1

VMS
VPN/Security Management System3

CHAPTER 1

1 Security Target Introduction

This Security Target (ST) describes the objectives, requirements and rationale for the Cisco Security Agent Version 4.5.1. The language used in this Security Target is consistent with the *Common Criteria for Information Technology Security Evaluation, Version 2.1*, the ISO/IEC JTC 1/SC27, *Guide for the Production of PPs and STs, Version 0.9* and all National Information Assurance Partnership (NIAP) and International Interpretations through June 16, 2004. As such, the spelling of terms is presented using the internationally accepted English.

1.1 Security Target Reference

This section provides identifying information for the Cisco Security Agent (CSA) Version 4.5.1 Security Target by defining the Target of Evaluation (TOE).

1.1.1 Security Target Name

Cisco Security Agent Version 4.5.1 Security Target
Revision 1
April 4, 2007

1.1.2 TOE Reference

Cisco Security Agent Version 4.5.1.655

1.1.3 Evaluation Assurance Level

Assurance claims conform to EAL2 (Evaluation Assurance Level 2) from the *Common Criteria for Information Technology Security Evaluation, Version 2.1*.

1.1.4 Keywords

Access Control, Attack, Intrusion, Intrusion Detection System, Agent

1.2 TOE Overview

This Security Target addresses the Cisco Security Agent Version 4.5.1 TOE. The TOE is a software-based intrusion detection and intrusion prevention application comprised of two essential components: the Management Center that installs on designated Windows systems and the Agent that installs on server and desktop Windows systems across the network. Functioning under specific policies to be defined by the needs of the deploying organization, the Management Center and Agent(s) work in parallel to defend against the proliferation of attempted intrusions and attack scenarios across networks and systems.

1.2.1 Security Target Organisation

Chapter 1 of this ST provides introductory and identifying information for the TOE.

Chapter 2 describes the TOE and provides some guidance on its use.

Chapter 3 provides a security environment description in terms of assumptions, threats and organisational security policies.

Chapter 4 identifies the security objectives of the TOE and of the Information Technology (IT) environment.

Chapter 5 provides the TOE security and functional requirements, as well as requirements on the IT environment.

Chapter 6 is the TOE Summary Specification, a description of the functions provided by the Cisco Security Agent Version 4.5.1 to satisfy the security functional and assurance requirements.

Chapter 7 identifies claims of conformance to a registered Protection Profile (PP).

Chapter 8 provides a rationale for the security objectives, requirements, TOE summary specification and PP claims.

1.3 Common Criteria Conformance

The Cisco Security Agent Version 4.5.1 is compliant with the Common Criteria (CC) Version 2.1, functional requirements (Part 2 extended) conformant, assurance requirements (Part 3) conformant for EAL2, and all National Information Assurance Partnership (NIAP) and International Interpretations through June 16, 2004.

1.4 Protection Profile Conformance

The Cisco Security Agent Version 4.5.1 does not claim conformance to any registered Protection Profile.

1.5 Document Conventions

The CC defines four operations on security functional requirements. The font conventions below identify the conventions for the operations defined by the CC.

Assignment: indicated with bold text

<u>Selection:</u> <u>indicated with underlined text</u>

Refinement: indicated with bold text and italics

Iteration: indicated with typical CC requirement naming followed by a number in parenthesis for each iteration (e.g., FMT_MOF.1 (1))

CHAPTER 2

2 TOE Description

This section describes the evaluation configuration of the CSA product. Also, it distinguishes the physical and logical boundaries of the TOE, highlights the assets and capabilities of the TOE, and defines the protection mechanisms and access rights to these assets.

2.1 Overview

Cisco Security Agent Version 4.5.1 is an agent-based intrusion detection and response system. CSA is designed to protect individual workstations and servers from a variety of application and network-based attacks. For example, CSA can be used to protect workstations in an office environment or a Web-server farm.

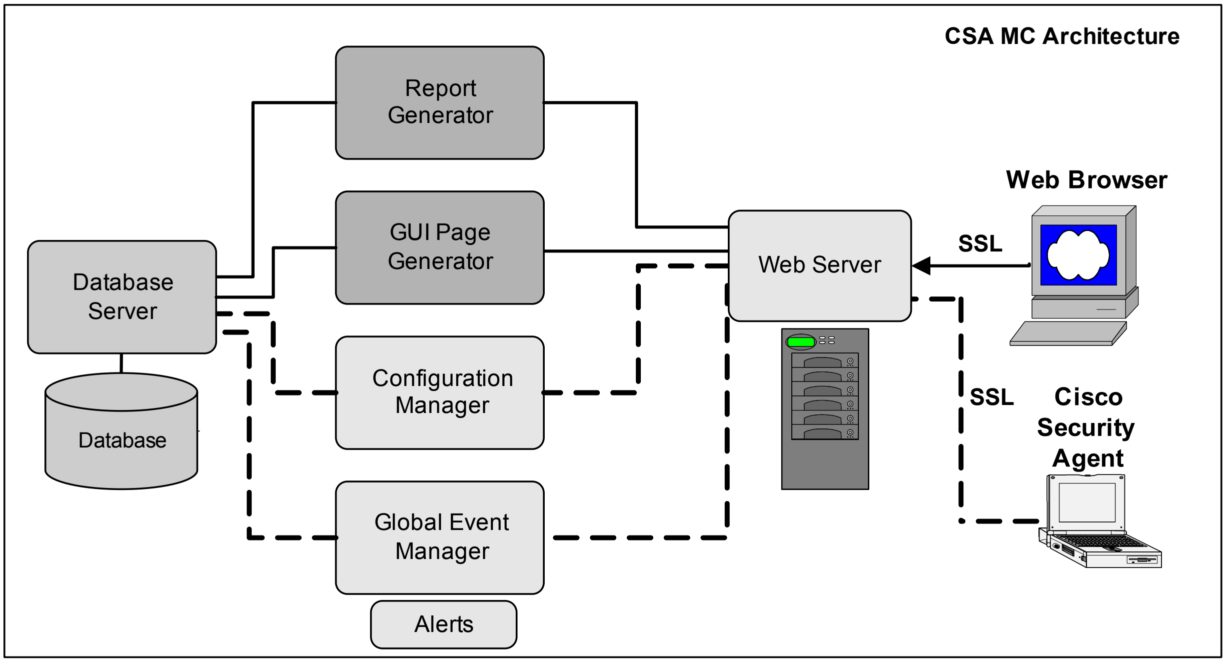
2.2 Architecture Description

CSA consists of a single Management Center (MC) for CSA and between one and 100,000 Agents. Both Host Agents and Server Agents are supported. The security functionality associated with them is identical, and this document refers to both types generically as Agents. The Management Center enables single-point administration of the Agents that are installed on desktops and servers throughout the network.

The CSA Management Center and an Agent, as represented in Figure 1, represent the TOE. CiscoWorks VPN/Security Management System (VMS) provides support infrastructure to the Management Center for CSA. It provides Identification & Authentication (I&A) functionality when administrators connect to the system, and performs session locking and re-authentication. VMS provides additional functionality to other Cisco management products, but this additional functionality is beyond the scope of this evaluation.

CSA Management Center (MC) architecture is displayed in Figure 1. Note that although the agent is mentioned often here, it is only in terms of CSA MC's relation to the agent. Agent software does have its own system components which are described in this chapter. It is CSA MC that pushes security policies to the agents and coordinates the events it receives back from the agents. The mechanisms that are required to perform those tasks are described here as part of the CSA MC architecture.

Figure 1-1 Management Center Security Architecture



The web browser, shown on the right in Figure 1-1, represents any web browser on any system across an enterprise from which administrators can securely access the CSA MC web-based interface. Communications between the web browser and the web server occur over SSL, allowing administrators to securely access the database of rule configurations from any location.

The web server provides the means of communication between the web browser and all other CSA MC system components. The web server displays reporting information, configuration version data, and event logging data.

It is through the web server that the agents installed on systems across an enterprise can exchange data with the CSA MC configuration manager and the global event manager. When agents poll in to CSA MC for rule set updates, it is the configuration manager that pulls the rules from the database and distributes them to the particular agents for which they are intended. Agents also send events to the global event manager which stores this information in the central SQL server database.

The SQL server database is the central repository for configuration data (host agents, groups, file rules, network rules, registry rules, etc.) created by the administrator and for the system event information provided by the agents. It is in this database that rules and information on system groupings are stored when the administrator generates rules and policies through the web-based interface. When reports are requested by the administrator, the report generator component gathers rule and event data kept in the database and produces reports using this information.

All information (rule configurations, event logs, etc.) passed between CSA MC and the agents distributed across your enterprise is encrypted providing a secure communication channel for the exchange of data.

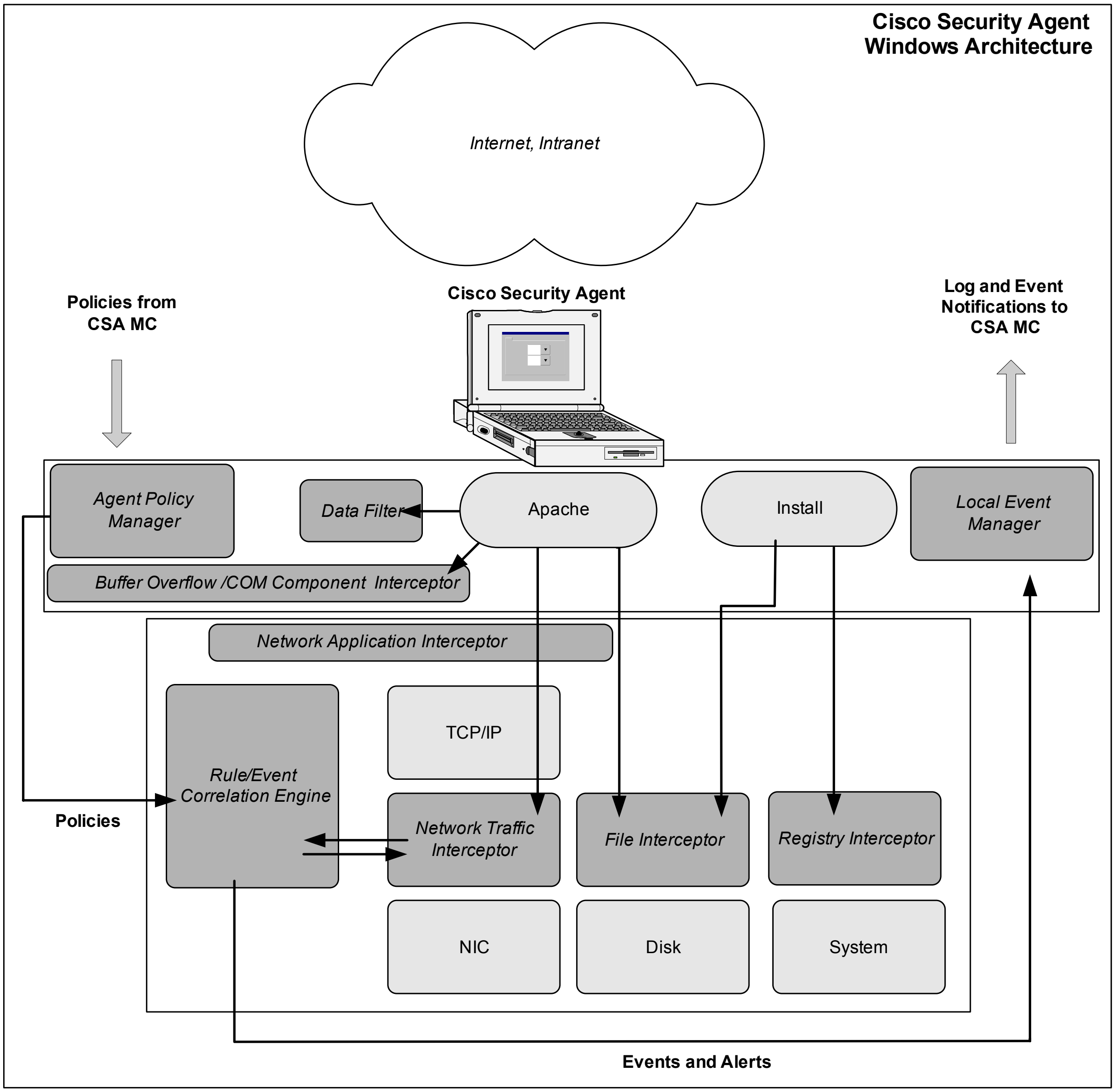
The TSF data of the Management Center are the Agent registrations, Agent grouping and policy configurations, the event logs, and a public/private-key certificate. These assets are stored in the database shown in Figure 1-1. The database infrastructure (binaries and raw data) is protected by the Agent on this host. Access to use the database is also restricted by the Agent to only the Management Center application. The binaries and configuration files for the Management Center are also valuable assets of the TOE. These binaries and configuration files are located within the CSA install directory and are protected by the Agent on this host.

The sensitive capabilities of the Management Center are the ability to publish Agent security policies and the ability to generate reports. These capabilities are only configurable from within the Web-based administration tool. Administration sessions are authenticated and use secure HTTP. The Web server's infrastructure (binaries and published resources) is protected by the Agent on this host. Access to administer the Web server is also restricted to only the Management Center application.

2.2.1 Agent Components

Figure 1-2 illustrates the security architecture of an Agent host in the TOE configuration. The dark shaded areas represent the TOE portion of the Agent host architecture. Figure 1-2 shows the agent in terms of its system components, displaying where those components operate in relation to general system functions. For example, the interceptors shown in the diagram install and work at the kernel level.

Figure 1-2 ***Cisco Security Agent Components (Windows)***



Starting from the left side of the diagram, the agent **policy manager** receives the rules configured by the administrator from CSA MC. These rules are sent to the agent's **rule/event correlation** engine. If a rule set already exists there, those rules are updated or replaced with the newest rule set.

The **interceptors** do as their name indicates, they intercept key actions that are attempted on the system and check the action in question against the rule correlation engine to determine if a rule set allows or denies it. Based on the information the interceptors receive, they either allow the action to take place or they stop it cold.

Actions are stopped based on certain criteria that are part of each rule and consequently each interceptor acts based on a component-targeted set of criteria.

For example, the **network application interceptor** controls which applications are allowed to communicate with the network, while the **network traffic interceptor** provides system hardening features such as SYN flood protection and port scan detection. The **file interceptor** controls which applications can read and/or write to specified system files and directories. The **registry interceptor** controls system behavior, preventing applications from writing to particular registry keys. All of these controls can be as broad or as granular as necessary.

As the interceptors are allowing or denying actions, they produce an event each time a rule set is triggered by a system action. These events are stored in the rule/event correlation engine which forwards them on to the **local event manager** and **global event manager**. Events are also stored in the NT event log or W2K event viewer on the agent system.

The sensitive assets of the Agent are the security policy, the events log, and the binaries of the Agent. The Agent always enforces a built-in self-protection policy as well as the explicit, downloaded policy. The built-in policy controls write access to all of the data files and binaries in the Agent install directory. This includes the events log, the security policy, and the binaries. The built-in policy also protects the DLLs and drivers that are in stored in appropriate directories of the operating system. This feature also protects write access to memory and disk space that is vital to Agent operation. The Agent on the Management Center host provides the same protection for the Management Center.

The sensitive capabilities of the Agent are the ability to receive security policies from the Management Center, the ability to send events to the Management Center, and the ability to enforce its security policy. The first two capabilities are communications between the Agents and the Management Center. These communications utilize the secure HTTP capability provided by the Web server to keep the data from being intercepted. In addition, policy enforcement is protected by the assumption that the operating system always invokes the TSF and provides dedicated process space for the TOE.

The following table presents the required hardware and software specifications of the non-TOE components given in Figure 1-1 and Figure 1-2.

Table 1-1 Platform Requirements of the IT Environment

Component	Description
Management Center Host	PC with 1GHz or faster processor CD-ROM drive 100Base-T or faster connection 1 GB RAM 9 GB available disk drive space 2 GB virtual memory Color monitor with video card capable of 16-bit color Windows 2000 Server or Advanced Server, Service Pack 4

Web Browser	Internet Explorer v. 6.0 with Service Pack 1 Supporting 128 bit encryption Cookies enabled, maximum medium setting for Internet Security JavaScript enabled.
Agent Host	Intel Pentium 200MHz or faster processor 128 MB system memory or greater 15 MB disk space or greater One Ethernet interface supporting TCP/IP Windows 2000 Professional, Server, or Advanced Server, Service Pack 0, 1, 2 or 3 -OR- Windows NT 4.0 Workstation, Server, or Enterprise Server, Service Pack 4 or higher -OR- Windows XP (Professional English 128 bit), Service Pack 0 or 1 -OR- Windows 2003

2.3 Physical Boundaries

The items listed in the following table comprise the TOE. The TOE does not include any component that is not specified below. Specifically, the TOE does not include any hardware, any operating system which TOE operates upon, any Web server, any Database, any network, or any applications running on the Agent host. The table is broken down into components for each installation type: the Management Center and the Agent.

Table 1-2 Physical Components

Installation	Physical Component
Management Center	Report Generator Web Application GUI Page Generator Configuration Manager Global Event Manager
Agent	Rule/Event Correlation Engine AgentPolicy Manager Local Event Manager Buffer Overflow/COM Component Interceptor File Interceptor Registry Interceptor Network Application Interceptor Network Traffic Interceptor

2.4 Logical Boundaries

The logical boundaries of the TOE include security features of the Agent and the secured interfaces to the Management Center.

The Agent includes many security features. Agents generate events and perform actions on those events. Defining an event and mapping action(s) to them is done by the Agent's security policy. The origin of this policy was discussed in the Architecture section, above.

Additionally, the Management Center supports its own policy of events and actions and provides security features related to system management. The following subsections preview the features available in the TOE, thus defining its logical boundary.

2.4.1 Security Audit

Records of Program Access Control Policy enforcement, malicious activity, and system management events are logged by the Agent into secured disk space on the Agent host. These event records are also sent to the Management Center. Additionally, when the Agent detects malicious activity, the Agent will intercept the offending process and prompt the End User for guidance.

2.4.2 Security Management

Robust security management capabilities exist in the CSA product. Within the Management Center, Agents can be assembled into groups, to which security policies can be attached. The Administrator configures the security policies in the Management Center over secure HTTP to a Web application. The Web server and Web application are part of the Management Center installation. These configurations are then deployed to the Agents via secure HTTP.

When an Agent is installed, it registers with the Management Center. At this time, the security policy is given to the Agent. Also, the Agent polls the Management Center at configurable intervals for policy updates.

2.4.3 User Data Protection

The TOE provides data protection by enforcing access control on files, process memory space, and the Windows Registry. Also, the TOE controls the ability for applications to execute. These features protect against email-worms, keystroke logging, code injection, and buffer overflows.

Information flow to and from the machine through the network interface is control by the TOE. This capability enables detection and/or protection from network scans, packet-sniffers, Syn-flood attacks, and malformed packet attacks.

2.5 Evaluated Configuration

The evaluated configuration includes the Management Center executing on a Windows platform (with VMS) and Agents executing on Windows hosts. The evaluated configuration does not include Server Agents for Solaris or Linux.

CHAPTER 3

3 TOE Security Environment

This chapter identifies assumptions (A), threats (T), and organisational security policies (P) related to the TOE. Assumptions are given to detail the expected environment and operating conditions of the system. Threats are those that are addressed by the TOE. And, organisational security policies are specific rules, procedures, or practices that are part of the TOE

3.1 Assumptions

The assumptions are ordered into three groups. They are personnel assumptions, physical environment assumptions, and IT environment assumptions. Personnel assumptions describe characteristics of personnel who are relevant to the system. Physical environment assumptions describe characteristics of the non-IT environment that the system is deployed in. In addition, IT environment assumptions describe the technology environment that the TOE is operating within.

3.1.1 Personnel Assumptions

A.NOEVILADMIN The Administrator is non-hostile and follows all administrator guidance when using the TOE.

A.NOEVILUSERS Authorized users of Agent hosts (End Users) are non-hostile and do not attempt to attack or subvert the CSA system and its policy.

A.PLATFORM_A The Administrator will install and configure the platforms protected by the Agents in conformance with Table 1.

A.PLATFORM_MC The Administrator will install and configure the platform used to host the Management Center in conformance with Table 1.

A.INSTALL The Administrator will install and configure the hardware, operating systems, and software required to support the TOE in conformance with the CSA installation guides.

3.1.2 Physical Environment Assumptions

A.ENVIRON_A The Agent will be located in an environment that provides physical security.

A.ENVIRON_MC The Management Center will be located in an environment that provides physical , uninterruptible power, air conditioning, and all other conditions required for reliable operation of the hardware.

3.1.3 IT Environment Assumptions

Table 1 presents the required hardware and software specifications of the IT environment.

3.2 Threats

The TOE or IT environment addresses the threats identified in the following sections.

3.2.1 Threats Addressed by the TOE

The TOE addresses the threats discussed below. The threat agents are either unauthorized persons or external IT entities not authorized to use the TOE itself.

T.KEYLOG A malicious program may be executed on a system protected by the TOE that attempts to monitor all keystrokes entered by an End User to gain password information or other sensitive data.

T.PORTSCAN An attacker may send network traffic that is received on a system protected by the TOE that attempts to scan ports as a means to gather information about or identify weaknesses in systems protected by the TOE.

T.SYNFLOOD An attacker may send network traffic that is received on a system protected by the TOE that attempts to SYN-flood a server system protected by the TOE. When a TCP/IP connection request is received from a return address that is not in use, a useless, half-open connection will persist for a period of time. Having too many of these connections will prevent legitimate connections from being established.

T.MALPACK An attacker may send network traffic that is received on a system protected by the TOE that attempts to exercise a bug in the operating system's network implementation. This type of attack can cause the system to crash.

T.OVERFLOW A program may be executing on a system protected by the TOE that reads data from the network which causes an overflow of memory buffers. If this happens the network data may contain and execute arbitrary code with full privilege on the system.

T. WORM A malicious email attachment may execute on a system protected by the TOE that attempts to send itself to other networked systems. The malicious execution may also modify Windows Registry keys, write its own script files or modify existing files on the system protected by the TOE.

T.TROJAN A malicious program may be executed on a system protected by the TOE that attempts to inject malicious code into the memory space of another process.

T.PWDTHEFT A malicious program may be executed on a system protected by the TOE that attempts to access a restricted area of the Windows Registry that contains the hashes of system passwords.

T.COVERT A malicious program may be executed on a system protected by the TOE that attempts to send data covertly over the network utilizing unsolicited ICMP response packets.

T.REGACC A malicious program may attempt to gain unauthorized access to the Windows Registry and disclose or corrupt sensitive information stored there.

T.FILEACC A malicious program may attempt to gain unauthorized access to the file system and disclose or corrupt sensitive information stored in files.

T.NETACC A malicious program may attempt to gain unauthorized access to network functions such as sending information, creating server sockets to receive information, setting the network interface to promiscuous mode, or sending ICMP packets for the purpose of subverting a host protected by the TOE.

T.COMACC A malicious program may attempt to gain unauthorized access to Component Object Model components in order to use their functions to carry out some part of an attack for the purpose of subverting a host protected by the TOE.

T.BYPASS A malicious subject on a platform protected by the TOE may access the TSF or TSF data without invoking the TSF.

3.2.2 Threats Addressed by the Operating Environment

TE.TAMPER A malicious subject may gain unauthorized access to TSF data.

TE.INTRCPT_A A malicious subject may intercept or modify unencrypted network traffic between the Management Center and the Agent for the purpose of subverting the TOE or a host protected by the TOE.

TE.INTRCPT_MC A malicious subject may intercept or modify unencrypted network traffic between the Management Center and the Administrator’s HTML browser for the purpose of subverting the TOE or a host protected by the TOE.

TE.UNAUTH	An attacker may attempt to assume the identity of the Administrator in order to modify the TOE configuration.

3.3 Organisational Security Policies

None

CHAPTER 4

4 Security Objectives

4.1 Security Objectives for the TOE

The following are the IT security objectives for the TOE:

O.SAFENET The TOE will monitor inbound network traffic to detect and mediate port-scans, Syn-flood attacks, and malformed network packets from going to the operating system.

O.SAFECODE The TOE will monitor processes to detect and mediate attempts at keystroke logging, buffer overflows, email worms, code injection, hashed password theft, and covert ICMP channels by untrusted entities.

O.REGCON The TOE will monitor attempts to access the Windows Registry and allow or deny access based on a set of access control rules, the Program Access Control Policy.

O.FILECON The TOE will monitor attempts to access the file system and allow or deny access based on the Program Access Control Policy.

O.NETCON The TOE will monitor outbound attempts to access the network and allow or deny access based on the Program Access Control Policy.

O.COMCON The TOE will monitor attempts to access Component Object Model components and allow or deny access based on the Program Access Control Policy.

O.AUDIT The TOE will keep an audit log of security-relevant events detected by the TOE and provide a mechanism for the Administrator to review those logs.

O.MANAGE The TOE will provide secure management of the Program Access Control Policy.

O.NOBYPASS The TOE will not allow an application or user to access TSF or TSF data without invoking the TSP enforcement functions.

4.2 Security Objectives for the Environment

The security objectives for the IT environment are listed below.

OE.DEDICATED The Management Center platform is dedicated to the Management Center software and serves no other purpose.

OE.COMSEC Communications between the Management Center and Agents and communications between the Management Center and the Administrator’s HTML browser will be performed using secure HTTP.

OE.TIMESTAMP All operating system platforms will provide a reliable source of time information.

OE.AUTH The IT Environment will identify and authenticate users before allowing them any access to the CSA administrative interface. Open sessions will timeout and provide session locking.

The non-IT security objectives listed below are to be satisfied without imposing technical requirements on the TOE. Thus, they will be satisfied through application of procedural or administrative measures.

OE.PLATFORM_A The Administrator will install and configure the platforms protected by the Agents in conformance with Table 1.

OE.PLATFORM_MC The Administrator will install and configure the platform used to host the Management Center in conformance with Table 1.

OE.INSTALL The Administrator will install and configure the hardware, operating systems, and software required to support the TOE in conformance with the CSA installation guides.

OE.ENVIRON_A The Agent will be located in an environment that provides physical security.

OE.ENVIRON_MC The Management Center will be located in an environment that provides physical security, uninterruptible power, air conditioning, and all other conditions required for reliable operation of the hardware.

OE.NOEVILADMIN The Administrator is not hostile, has been trained in the use of the TOE, and follows all administrator guidance.

OE.NOEVILUSERS Authorized users of hosts protected by the TOE are not hostile, have been trained in the use of the TOE, and follow guidance

4.3 Rationale for IT Security Objectives

This section provides the rationale that all IT security objectives address threats against the TOE or the Environment.

O.SAFENET This objective addresses the threats from malicious inbound network connections, including T.PORTSCAN, T.SYNFLOOD, and T.MALPACK.

O.SAFECODE This objective addresses the threats from malicious code that runs on the host protected by an Agent, including T.OVERFLOW, T.WORM, T.KEYLOG, T.TROJAN, T.PWDTHEFT, and T.COVERT.

O.REGCON This objective addresses the threat from a program attempting unauthorized access to the Windows Registry, or T.REGACC.

O.FILECON This objective addresses the threat from a program attempting unauthorized access to the file system, or T.FILEACC.

O.NETCON This objective addresses the threat from a program attempting unauthorized access to the network, or T.NETACC.

O.COMCON This objective addresses the threat from a program attempting unauthorized access to a Component Object Model (COM) component, or T.COMACC.

O.AUDIT This objective addresses all threats countered by the TOE. By requiring the TOE to record security-relevant events and provide the Administrator with review capabilities the TOE enables the Administrator to detect malicious activity and verify proper system behaviour.

O.MANAGE This objective addresses T.REGACC, T.FILEACC, T.NETACC, T.COMACC by requiring the TOE to provide management of the Program Access Control Policy. The Program Access Control Policy is enforced to directly address the threats listed above.

O.NOBYPASS This objective covers the threat T.BYPASS. By requiring the TSP enforcement functions to be invoked before access to the TSF and TSF data is allowed, the threat is directly mitigated.

The objectives below are levied on the environment.

OE.DEDICATED This objective covers the threat TE.TAMPER. By requiring the Management Center platform to be dedicated to the TOE, no other processes will be on that platform that may interfere with the TSF.

OE.COMSEC This objective covers TE.INTRCPT_A and TE.INTRCPT_MC, the threats of TOE communications being disclosed or modified during transmission. These are covered because OE.COMSEC requires the use of secure HTTP for these communications.

OE.TIMESTAMP This objective addresses all threats countered by the TOE. By providing the TOE with a reliable timestamp, OE.TIMESTAMP supports the TOE's auditing capabilities. Auditing is used by the TOE to counter all of its threats.

OE.AUTH This objective addresses the threat of allowing unauthorized access to the CSA administrative functions.

Table 1-3 Mappings for IT Security Objectives to Threats

	T.PORTSCAN	T.SYNFLOOD	T.MALPACK	T.OVERFLOW	T.WORM	T.KEYLOG	T.TROJAN	T.PWDTHEFT	T.COVERT	T.REGACC	T.FILEACC	T.NETACC	T.COMACC	T.BYPASS	TE.UNAUTH	TE.TAMPER	TE.INTRCPT_A	TE.INTRCPT_MC
O.SAFENET	X	X	X															
O.SAFECODE				X	X	X	X	X	X									
O.REGCON										X								
O.FILECON											X							
O.NETCON												X						
O.COMCON													X					
O.AUDIT	X	X	X	X	X	X	X	X	X	X	X	X	X					
O.MANAGE										X	X	X	X					
O.NOBYPASS														X				
OE.DEDICATED																X		
OE.COMSEC																	X	X
OE.TIMESTAMP	X	X	X	X	X	X	X	X	X	X	X	X	X					
OE.AUTH															X			

4.4 Rationale for Non-IT Security Objectives for the Environment

This section provides the rationale that all non-IT security objectives for the environment address threats or assumptions.

OE.NOEVILADMIN This objective addresses A.NOEVILADMIN by restating the assumption.

OE.NOEVILUSERS This objective addresses A.NOEVILUSERS by restating the assumption.

OE.PLATFORM_A This objective addresses A.PLATFORM_A by restating the assumption.

OE.PLATFORM_MCThis objective addresses A.PLATFORM_MC by restating the assumption.

OE.INSTALL This objective addresses A.INSTALL by restating the assumption.

OE.ENVIRON_A This objective addresses A.ENVIRON_A by restating the assumption.

OE.ENVIRON_MC This objective addresses A.ENVIRON_MC by restating the assumption.

Table 1-4 Mappings for Assumptions to Security Objectives for the Environment

	A.NOEVILADMIN	A.NOEVILUSERS	A.PLATFORM_A	A.PLATFORM_MC	A.INSTALL	A.ENVIRON_A	A.ENVIRON_MC
OE.NOEVILADMIN	X						
OE.NOEVILUSERS		X					
OE.PLATFORM_A			X				
OE.PLATFORM_MC				X			
OE.INSTALL					X		
OE.ENVIRON_A						X	
OE.ENVIRON_MC							X

4.5 Rationale for Threat Coverage

This section provides a justification that for each threat, the security objectives counter the threat.

T.PORTSCAN Port-scans can be detected by watching for a pattern of connection requests. O.SAFENET detects port-scans. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T.SYNFLOOD Syn-flood attacks can be detected by watching for a pattern of half-open connection requests. O.SAFENET detects Syn-floods. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T.MALPACK Malicious packets can be detected by watching for packets that do not follow the specification for their protocol. O.SAFENET detects malformed packets. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T.OVERFLOW Buffer overflows can be detected by checking whether return addresses reference memory in the stack or heap. O.SAFECODE detects buffer overflows. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T. WORM

Email worms can be detected by monitoring the activities of programs downloaded from email and the network. O.SAFECODE detects email worms. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T.KEYLOG

Keystroke logging can be detected by monitoring requests to register for keystroke events from other processes. O.SAFECODE detects keystroke logging. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T.TROJAN

Code injection can be detected by monitoring the activities of applications. O.SAFECODE requires the TOE to monitor untrusted processes for code injection. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T.PWDTHEFT

Access to the restricted area of the Windows Registry that stores hashed system passwords must be detected according to O.SAFECODE. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T.COVERT

Unsolicited ICMP response packets must be detected according to O.SAFECODE. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T.REGACC

O.REGCON addresses the threat of unauthorized access to the Windows Registry by enforcing an access control policy upon the Registry. O.AUDIT addresses this threat by security relevant events related to the registry. O.MANAGE addresses this threat by providing a mechanism to configure this access control policy. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T.FILEACC

O.FILECON addresses the threat of unauthorized access to the file system by enforcing an access control policy upon the file system. O.AUDIT addresses this threat by security relevant events related to the file system. O.MANAGE addresses this threat by providing a mechanism to configure this access control policy. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T.NETACC

O.NETCON addresses the threat of unauthorized outbound access to networking functions by enforcing an access control policy upon networking functions. O.AUDIT addresses this threat by security relevant events related to the network. O.MANAGE addresses this threat by providing a mechanism to configure this access control policy. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T.COMACC O.COMCON addresses the threat of unauthorized access to Component Object Model components by enforcing an access control policy upon COM components. O.AUDIT addresses this threat by security relevant events related to COM components. O.MANAGE addresses this threat by providing a mechanism to configure this access control policy. Also, O.AUDIT and OE.TIMESTAMP mitigate this threat by providing a security event audit trail.

T.BYPASS O.NOBYPASS addresses this threat of untrusted subjects utilizing the TSF without invoking the TSP enforcement functions by directly forbidding it.

The threats below are against the environment.

TE.TAMPER OE.DEDICATED addresses this threat of another process being able to tamper with the execution of the TSF.

TE.INTRCPT_A OE.COMSEC addresses this threat of untrusted subjects intercepting unencrypted inner-TOE communications. This would include communications between the Agents and the Management Center.

TE.INTRCPT_MC OE.COMSEC addresses this threat of untrusted subjects intercepting unencrypted inner-TOE communications. This would include communications between the Administrator’s HTML browser and the Management Center.

TE.UNAUTH OE.AUTH addresses this threat by requiring the IT Environment to identify and authenticate the Administrator before allowing administrative access and by locking idle sessions that increase the risk of unauthorised access to the TOE.

CHAPTER 5

5 IT Security Requirements

This section contains the security requirements that are relevant to the TOE. These requirements consist of functional components from Part 2 of the CC as well as explicitly stated requirements, and assurance components from Part 3 of the CC. Any SFR that is marked up by ***-NIAP-XXXX***, is to be considered an explicitly stated requirement. These SFRs correspond with SFRs in the Common Criteria for which a National Information Assurance Partnership (NIAP) interpretation exists.

This section also contains the Strength of Function claim and corresponding rationale for components that require such a claim.

Table 1-5 Security Functional Requirements

TOE Security Functional Requirements	
FAU_ARP.1	Security alarms
FAU_GEN.1***-NIAP-0347***	Audit data generation
FAU_SAA.3	Simple Attack Heuristics
FAU_SAR.1	Audit Review
FAU_SAR.3	Selectable Audit Review
FAU_SEL.1***-NIAP-0407***	Selective Audit
FAU_STG.1***-NIAP-0422***	Protected Audit Trail Storage
FDP_ACC.1	Subset Access Control
FDP_ACF.1***-NIAP-0407***	Security Attribute Based Access Control
FMT_MOF.1	Management of Security Functions Behaviour
FMT_MSA.1	Management of Security Attributes
FMT_MSA.3	Static Attribute Initialisation
FMT_MTD.1 (1)	Management of TSF Data
FMT_MTD.1 (2)	Management of TSF Data
FMT_SMF.1	Specification of Management Functions
FMT_SMR.1	Security Roles
FPT_RVM.1	Non-Bypassability of the TSP
IT Environment Security Functional Requirements	
FIA_UAU.2	User Authentication Before Any Action
FIA_UAU.6	Re-Authenticating
FIA_UID.2	User Identification Before any Action
FPT_ITT.1	Basic Internal TSF Data Transfer Protection
FPT_SEP.1	TSF Domain Separation
FPT_STM.1	Reliable Time Stamps
FTA_SSL.2	User-Initiated Locking
FTP_TRP.1	Trusted Path

5.1 TOE Security Functional Requirements

The TOE security functional requirements for this Security Target consist of the following components from Part 2 of the CC.

5.1.1 Security Audit (FAU)

5.1.1.1 FAU_ARP.1 Security Alarms

FAU_ARP.1.1 The TSF shall log, deny, or **query the End User to Allow or Deny the offending signature (per table 6)** upon detection of a potential security violation.

5.1.1.2 FAU_SAA.3 Simple Attack Heuristics

FAU_SAA.3.1 The TSF shall be able to maintain an internal representation of the following signature events **listed in the following table** that may indicate a violation of the TSP.

FAU_SAA.3.2 The TSF shall be able to compare the signature events against the record of system activity discernible from an examination of **the supporting data listed in the following table.**

FAU_SAA.3.3 The TSF shall be able to indicate an imminent violation of the TSP when a system event is found to match a signature event that indicates a potential violation of the TSP.

Table 1-6 Signature Events and Their Respective Data

Signature Events	Supporting Data	Default Response
Programs that attempt to sniff network traffic	-System calls attempting to set promiscuous mode on the network interface -System calls attempting to register non-IP protocols -List of downloaded programs	Log. Event is detected by the network interface, so no option to query. Event is logged by the TSF.
IP port-scans and ICMP scans	-Network traffic	Deny. Event is detected by the network interface, so no option to query.
Syn-flood attacks	-Network traffic -List of known good external IP addresses	Deny. Event is detected by the network interface, so no option to query.
Malformed packets	-Network traffic	Deny. Event is detected by the network interface, so no option to query.
Programs that attempt to replicate themselves	-System calls attempting to use network interfaces -List of downloaded programs	Query. The rule is a default rule with the TOE and is defaulted to query the user.

Programs that attempt to log keystrokes	-System calls attempting to register for keystrokes -List of downloaded programs -The foreground/background status of each process	Query. The rule is a default rule with the TOE and is defaulted to query the user.
Programs that attempt to inject code into other processes	-System calls attempting to access another process' memory space -System calls attempting to register a call-back method with another process' events -List of downloaded programs	Query. The rule is a default rule with the TOE and is defaulted to query the user.
Programs that cause buffer overflows	-Return addresses for processes -Process memory space boundaries -List of programs that have accessed the network	Query. The rule is a default rule with the TOE and is defaulted to query the user.
Programs that steal hashed system passwords	-System calls attempting to gain access to the hashed passwords in the Windows Registry -List of downloaded programs	Query. The rule is a default rule with the TOE and is defaulted to query the user.
Execution of Mobile code	-Time -Attempts to execute new processes -Filename of executables used to start new processes -List of downloaded programs	Query. The rule is a default rule with the TOE and is defaulted to query the user.
Execution of Downloaded ActiveX controls	-List of Active X components -Attempts to execute ActiveX components	Query. The rule is a default rule with the TOE and is defaulted to query the user.
Use of ICMP covert channels	-Calls attempting to send (unsolicited) ICMP response packets -List of ICMP requests	Deny. Event is detected by the network interface, so no option to query.

5.1.1.3 FAU_SAR.1 Audit Review

FAU_SAR.1.1 The TSF shall provide **the Administrator** with the capability to read **date and time of events, type of event, subject identity, outcome of the event, severity level of event, object identity, and host identity** 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 Selectable Audit Review

FAU_SAR.3.1 The TSF shall provide the ability to perform searches, sorting, ordering of audit data based on **type of event, time, minimum severity, maximum severity, or specified host**.

5.1.2 User Data Protection (FDP)

5.1.2.1 FDP_ACC.1 Subset Access Control

FDP_ACC.1.1 The TSF shall enforce the **Program Access Control Policy** on **subjects accessing files, network connections, the Windows Registry, and COM components**.

5.1.3 Security Management (FMT)

5.1.3.1 FMT_MOF.1 Management of Security Functions Behaviour

FMT_MOF.1.1 The TSF shall restrict the ability to determine the behaviour of, disable, enable, modify the behaviour of the functions **Program Access Control Policy** to **the Administrator**.

5.1.3.2 FMT_MSA.1 Management of Security Attributes

FMT_MSA.1.1 The TSF shall enforce the **Program Access Control Policy** to restrict the ability to modify the security attributes **the Program Access Control Policy Rules** to **the Administrator**.

5.1.3.3 FMT_MSA.3 Static Attribute Initialisation

FMT_MSA.3.1 The TSF shall enforce the **Program Access Control Policy** to provide restrictive default values for the security attributes that are used to enforce the SFP.

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

5.1.3.4 FMT_MTD.1 Management of TSF Data (1)

FMT_MTD.1.1 (1) The TSF shall restrict the ability to create, modify, delete the **Program Access Control Policy Rules** to **the Administrator**.

5.1.3.5 FMT_MTD.1 Management of TSF Data (2)

FMT_MTD.1.1 (2) The TSF shall restrict the ability to delete the **audit records** to **the Administrator**.

5.1.3.6 FMT_SMF.1 Specification of Management Functions

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

-create, modify, or delete the Program Access Control Policy Rules

-query, or delete the audit records.

5.1.3.7 FMT_SMR.1 Security Roles

FMT_SMR.1.1 The TSF shall maintain the roles **Administrator**.

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

5.1.4 Protection of the TSF (FPT)

5.1.4.1 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.

5.2 Explicitly Stated TOE Security Functional Requirements

The security functional requirements detailed in this section are explicitly stated requirements that identify security functional requirements of the TOE that are not currently defined in Part 2 of the CC.

5.2.1 Security Audit (FAU)

5.2.1.1 FAU_GEN.1-*NIAP-0347* Audit Data Generation

FAU_GEN.1.1-***NIAP-0347***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 **basic** level of audit; and

c) all actions resulting from the Program Access Control Policy.

FAU_GEN.1.2-***NIAP-0347***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) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, **the event severity level, the object identity, and host identity**.

5.2.1.2 FA U_SEL.1-*NIAP-0407* Selective Audit

FAU_SEL.1.1-***NIAP-0407*** The TSF shall be able to include or exclude auditable events from the set of audited events based on the following attributes:

a)object identity, subject identity, host identity, event type.

b)**time interval, severity level**.

5.2.1.3 FAU_STG.1-*NIAP-0422* Protected Audit Trail Storage

FAU_STG.1.1-***NIAP-0422***The TSF shall protect the stored audit records **in the audit trail** from unauthorised deletion.

FAU_STG.1.2-***NIAP-0422***The TSF shall be able to prevent modifications to the audit records **in the audit trail.**

5.2.2 User Data Protection (FDP)

5.2.2.1 FDP_ACF.1-*NIAP-0407* Security Attribute Based Access Control

FDP_ACF.1.1-***NIAP-0407***The TSF shall enforce the **Program Access Control Policy** to objects based on **the filename of the executable, version of the executable, the operation invoked, and the following attributes per type of access:**

-file access

ofile path

-network access

otarget address

oport

o**direction (client/server)**

-Windows Registry access

o**Registry key**

-COM component access

o**COM component name**

FDP_ACF.1.2-***NIAP-0407*** The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: [1]

[1] The precedence of the actions defined in FDP_ACF.1.2-4-***NIAP-0407*** is also conveyed in a table in the Access Control section of the TSS.

-Disallow the access, if the subject and object attributes match a Program Access Control Policy Rule (Policy Rule[2]) that specifies "High Priority Deny" or "High Priority Terminate."

-Allow the access, if the subject and object attributes match a Policy Rule that specifies "Allow" and do not match a Policy Rule that specifies "High Priority Deny" Or "High Priority Terminate."

-Query the user with a default of allow, if the subject and object attributes match a Policy Rule that specifies "Query the User Default Allow" and do not match a Policy Rule that specifies "High Priority Deny," "High Priority Terminate," or "Allow."

-Query the user with a default of deny, if the subject and object attributes match a Policy Rule that specifies "Query the User Default Deny" and do not match a Policy Rule that specifies "High Priority Deny," "High Priority Terminate", "Allow," or "Query the User Default Allow."

-Disallow the access, if the subject and object attributes match a Policy Rule that specifies "Deny" or "Terminate" and do not match a Policy Rule that specifies "High Priority Deny," "High Priority Terminate," "Allow," "Query the User Default Allow," or "Query the User Default Deny."

FDP_ACF.1.3-***NIAP-0407***The TSF shall explicitly authorise access of subjects to objects based on the following additional rules:

-**If the subject and object attributes do not match a Policy Rule**.

FDP_ACF.1.4-***NIAP-0407*** The TSF shall explicitly deny access of subjects to objects based on the following rules:

-no additional explicit denial rules.

5.3 TOE Security Assurance Requirements

The TOE meets the assurance requirements for EAL2 as defined by Part 3 of the CC. These assurance requirements are summarized in the following table.

[2] Policy Rules are composed by the Administrator from the subject and object attributes defined in FDP_ACF.1.1-***NIAP-0407***, following the guidelines and syntax that is defined in the document, *CSA Configuration Guide*.

Table 1-7 Assurance Requirements

Assurance Class	Component ID	Component Title
Configuration Management	ACM_CAP.2	Configuration Items
Delivery and Operation	ADO_DEL.1	Delivery Procedures
Delivery and Operation	ADO_IGS.1	Installation, Generation, and Start-Up Procedures
Development	ADV_FSP.1	Informal Functional Specification
Development	ADV_HLD.1	Descriptive High-Level Design
Development	ADV_RCR.1	Informal Correspondence Demonstration
Guidance Documents	AGD_ADM.1	Administrator Guidance
Guidance Documents	AGD_USR.1	User Guidance
Tests	ATE_COV.1	Evidence of Coverage
Tests	ATE_FUN.1	Functional Testing
Tests	ATE_IND.2	Independent Testing - Sample
Vulnerability Assessment	AVA_SOF.1	Strength of TOE Security Function Evaluation
Vulnerability Assessment	AVA_VLA.1	Developer Vulnerability Analysis

5.4 Security Requirements for the IT Environment

The security functional requirements for the IT Environment consist of the following components from Part 2 of the CC.

5.4.1 Identification and Authentication (FIA)

5.4.1.1 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 behalf of that user.

5.4.1.2 FIA_UAU.6 Re-Authenticating

FIA_UAU.6.1 The ***IT Environment*** shall re-authenticate the user under the conditions **after a period of inactivity specified by the Administrator between one and sixty minutes with default of fifteen minutes**.

5.4.1.3 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 behalf of that user.

5.4.2 Protection of the TSF (FPT)

5.4.2.1 FPT_ITT.1 Basic Internal TSF Data Transfer Protection

FPT_ITT.1.1 The ***IT environment*** shall protect TSF data from disclosure, modification when it is transmitted between separate parts of the TOE.

5.4.2.2 FPT_SEP.1 TSF Domain Separation

FPT_SEP.1.1 The ***IT Environment*** shall maintain a security domain for ***TSF*** 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.

5.4.2.3 FPT_STM.1 Reliable Time Stamps

FPT_STM.1.1 The ***IT environment*** shall be able to provide reliable time-stamps for ***use by the TOE***.

5.4.3 TOE Access (FTA)

5.4.3.1 FTA_SSL.2 User-Initiated Locking

FTA_SSL.2.1 The ***IT environment*** shall allow user-initiated locking of the user's own interactive session, by:

a)clearing or overwriting display devices, making the current contents unreadable;

b)disabling any activity of the user's data access/display devices other than unlocking the session.

FTA_SSL.2.2 The ***IT environment*** shall require the following events to occur prior to unlocking the session: **re-authentication**.

5.4.4 Trusted Path/Channels (FTP)

5.4.4.1 FTP_TRP.1 Trusted Path

FTP_TRP.1.1 The ***IT environment*** shall provide a communication path between ***the TOE*** and remote users that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure.

FTP_TRP.1.2 The ***IT environment*** shall permit <u>remote users</u> to initiate communication via the trusted path.

FTP_TRP.1.3 The ***IT environment*** shall require the use of the trusted path for <u>initial user authentication</u> <u>**and remote administration**</u>.

5.5 TOE Strength of Function Claim

The claimed minimum strength of function is SOF-basic. None of the TOE security functional requirements contain a permutational or probabilistic function.

5.6 Rationale for TOE Security Functional Requirements

The following table contains a mapping of the functional requirements and the security objectives each requirement enforces.

Table 1-8 Mapping of Functional Requirements to Objectives for the TOE

	O.SAFENET	O.SAFECODE	O. REGCON	O. FILECON	O.NETCON	O.COMCON	O. AUDIT	O. MANAGE	O. NOBYPASS
FAU_ARP.1	X	X							
FAU_GEN.1-*NIAP-0347*							X		
FAU_SAA.3	X	X							
FAU_SAR.1							X		
FAU_SAR.3							X		
FAU_SEL.1-*NIAP-0407*							X		
FAU_STG.1-*NIAP-0422*							X		
FDP_ACC.1			X	X	X	X			
FDP_ACF.1-*NIAP-0407*			X	X	X	X			
FMT_MOF.1								X	
FMT_MSA.1								X	
FMT_MSA.3								X	
FMT_MTD.1 (1)								X	
FMT_MTD.1 (2)							X		
FMT_SMF.1							X	X	
FMT_SMR.1								X	
FPT_RVM.1									X

This section provides the rationale for the functional requirements, including the security objectives enforced by the functional requirements.

FAU_ARP.1 Security alarms help to meet the objectives for detecting malicious behaviour in O.SAFENET and O.SAFECODE.

FAU_GEN.1-***NIAP-00347*** Audit data generation supports O.AUDIT.

FAU_SAA.3 Attack heuristics are used to detect attacks from the network and from downloaded code. This behaviour supports O.SAFENET and O.SAFECODE.

FAU_SAR.1 Audit review supports the O.AUDIT objective.

FAU_SAR.3 Selectable audit review supports the O.AUDIT objective.

FAU_SEL.1-***NIAP-0407*** Selective audit supports the O.AUDIT objective.

FAU_STG.1-***NIAP-0422*** Protected audit trail storage supports the O.AUDIT objective.

FDP_ACC.1 Subset access control is required to support O.FILECON, O.NETCON, O.COMCON, and O.REGCON.

FDP_ACF.1-***NIAP-0407*** Security attribute based access control is required to support O.FILECON, O.NETCON, O.COMCON, and O.REGCON.

FMT_MOF.1 Management of security functions behaviour is required to support O.MANAGE.

FMT_MSA.1 Management of security attributes is required to support O.MANAGE.

FMT_MSA.3 Static attribute initialization is required to support O.MANAGE.

FMT_MTD.1 (1) Management of Program Access Control Policy Rules is required to support O.MANAGE.

FMT_MTD.1 (2) Management of audit data is required to support O.AUDIT.

FMT_SMF.1 Management of Program Access Control Policy Rules is required to support O.MANAGE. Management of audit data is required to support O.AUDIT.

FMT_SMR.1 Security roles are required to support O.MANAGE.

FPT_RVM.1 Non-Bypassability of the TSP directly supports O.NOBYPASS.

5.7 Rationale for TOE Security Assurance Requirements

EAL2 was chosen to provide a moderate level of independently assured security. The chosen assurance level is consistent with the threat environment. Specifically, that the threat of malicious attacks is not greater than moderate and the product will have undergone a search for obvious flaws.

The TOE stresses assurance through vendor actions that are within the bounds of current best commercial practice. The TOE provides, primarily via review of vendor-supplied evidence, independent confirmation that these actions have been competently performed.

The general level of assurance for the TOE is:

A) Consistent with current best commercial practice for IT development and provides a product that is competitive against non-evaluated products with respect to functionality, performance, cost, and time-to-market.

B) The TOE assurance also meets current constraints on widespread acceptance, by expressing its claims against EAL2 from part 3 of the Common Criteria.

5.8 Rationale for IT Environment Security Requirements

The following table contains a mapping of the IT environment security functional requirements and the security objectives each requirement enforces.

Table 1-9 Mappings Between Functional Requirements and IT Security Objectives for the Environment

	OE.DEDICATED	OE.COMSEC	OE.TIMESTAMP	OE.AUTH
FIA_UAU.2				X
FIA_UAU.6				X
FIA_UID.2				X
FPT_ITT.1		X		
FPT_SEP.1	X			
FPT_STM.1			X	
FTA_SSL.2				X
FTP_TRP.1		X		

This section lists the IT objectives levied on the environment and the functional requirements that support each objective.

OE.AUTH Supported by FIA_UAU.2, FIA_UAU.6, FIA_UID.2, and FTA_SSL.2, because these SFRs ensure administrative users identify and authenticate themselves before gaining access to the administrative functions.

OE.DEDICATED Supported by FPT_SEP.1, because the SFR no untrusted subjects can tamper with the execution of the TSF.

OE.COMSEC Supported by FPT_ITT.1 and FTP_TRP.1, because these SFRs provide confidentiality and integrity for communications from the

Management Center to Agents and from the Management Center to the Administrator’s HTML browser, respectively. Protecting both these channels of communication was required by the objective.

OE.TIMESTAMP Supported by FPT_STM.1, because the SFR provides reliable timestamps as required by the objective.

5.9 Rationale for IT Security Requirement Dependencies

The following table lists the claimed TOE and IT Environment security requirements and their dependencies. This section also contains a rationale for any dependencies that are not satisfied. For the purpose of dependencies, SFRs with NIAP or International Interpretations are considered to fulfill the dependency of their original SFR, as interpretations do not alter the scope of the SFR.

Table 1-10 Functional Requirements Dependencies

SFR	Dependencies	Hierarchical To
FAU_ARP.1	FAU_SAA.1	None
FAU_GEN.1***-NIAP-0347***	FPT_STM.1	None
FAU_SAA.3	None	FAU_SAA.1
FAU_SAR.1	FAU_GEN.1	None
FAU_SAR.3	FAU_SAR.1	None
FAU_SEL.1***-NIAP-0407***	FAU_GEN.1 FMT_MTD.1	None
FAU_STG.1***-NIAP-0422***	FAU_GEN.1	None
FDP_ACC.1	FDP_ACF.1	None
FDP_ACF.1***-NIAP-0407***	FDP_ACC.1 FMT_MSA.3	None
FIA_UAU.2	FIA_UID.1	FIA_UAU.1
FIA_UAU.6	None	None
FIA_UID.2	None	FIA_UID.1
FMT_MOF.1	FMT_SMF.1 FMT_SMR.1	None
FMT_MSA.1	[FDP_ACC.1 or FDP_IFC.1] FMT_SMF.1 FMT_SMR.1	None
FMT_MSA.3	FMT_MSA.1 FMT_SMR.1	None
FMT_MTD.1	FMT_SMF.1 FMT_SMR.1	None
FMT_SMF.1	None	None
FMT_SMR.1	FIA_UID.1	None
FPT_ITT.1	None	None
FPT_RVM.1	None	None
FPT_SEP.1	None	None
FPT_STM.1	None	None
FTA_SSL.2	FIA_UAU.1	None
FTP_TRP.1	None	None

FAU_ARP.1 depends on FAU_SAA.1. Since FAU_SAA.3 is hierarchical to FAU_SAA.1, this dependency is satisfied.

FAU_SAR.1, FAU_SEL.1-***NIAP-0407*** and FAU_STG.1-***NIAP-0422*** depend on FAU_GEN.1. FAU_GEN.1-***NIAP-0347*** satisfies these dependencies.

FAU_SEL.1-***NIAP-0407*** depends on FMT_MTD.1. FMT_MTD.1 (2) satisfies this dependency.

FDP_ACC.1 depends on FDP_ACF.1. FDP_ACF.1-***NIAP-0407*** satisfies this dependency.

FTA_SSL.2 depends on FIA_UAU.1. Since FIA_UAU.2 is hierarchical to FIA_UAU.1, this dependency is satisfied.

FIA_UAU.2 and FMT_SMR.1 depend on FIA_UID.1. Since FIA_UID.2 is hierarchical to FIA_UID.1, these dependencies are satisfied.

CHAPTER 6

6 TOE Summary Specification

6.1 TOE Security Functions

This section describes the security functions implemented by the TOE to meet the TOE SFRs.

6.1.1 Audit

Every action resulting from the Agent's enforcement of the Program Access Control Policy results in the generation of an event record. Event records are also generated for the start-up and shutdown of auditing and changes to the Program Access Control Policy Rules. Event records document these system events according to designated time frames, event severity levels, and the system that generated the event. These event records are sent to one or more event sinks depending on the event. The event sinks include: the Management Center, which stores the event record in the events database for reporting and monitoring purposes; the Windows NT/2000 operating system event log, where the event record is stored in the application event log, this allows the local OS administrator to troubleshoot why certain access is failing, and the Agent user interface, where the event record is displayed in the messages pane.

Events are sent to one or more sinks based on specified criteria. Event records that are not related to processes accessing network resources are sent to the Management Center and the Windows NT/2000 event log but are not displayed in the messages pane of the Agent user interface. Deny actions on network access operations, for example, result in events being written to the Windows NT/2000 event log, the Management Center, and the Agent user interface.

Auditing on the Management Center (MC) can be selective. The MC can audit events that it receives from the IAs based on object identity, subject identity, host identity, event type, time interval, and severity level. This is needed to keep the size of logs on the MC to a manageable size. The Administrator can also purge audit records.

The Management Center supports several methods for accessing the information recorded in the events database. Various administrative Management Center views show the status of the Agent machines based on the event records received. These views provide a summary of the records including the option for detailed real-time views. All reports are displayed using one of two methods: a plain HTML view that may be viewed in any browser, or one that requires an ActiveX viewer to be installed on the browser. A reporting function enables the generation of hard copy reports in various formats for the purposes of historical viewing and analysis of events.

All Management Center reporting functions have a data selection component that determines which data is to be included in the report based on criteria specified by the Administrator. The Administrator can search, sort, or order event records into a report. Selection criteria for displaying event records is based on whether a record matches a specified event set or whether the event matches a filter that defines a start and end time, a minimum and maximum severity level, and a specific host of interest.

All event log reports select event records that are to be included in the report based on matching the event records against event sets specified by the Administrator. The event log records the specific event type, time, the host that generated the event, the severity level and the action taken. This information is included in the log report when specified by the Administrator.

A report based on event severity level is segmented based on the severity attribute of the event record. Within each section, records are displayed in sorted order. The Administrator specifies the sort order based on the time stamp, source host or event code attributes of the event record.

A report based on group membership is sectioned according to the group membership of the host that generated the event. Within each section, records are displayed in sorted order. The order is determined by the Administrator based on the time stamp, severity, or event code attributes of the event record.

6.1.2 Monitoring & Detection

The Agent intercepts accesses to assets protected by the TOE at the operating system level to verify their authority. This includes attempts to access the TSF and TSF data like audit records, attack signatures, and access control rights. Only authorized accesses through the TSF are permitted. This monitoring also provides the Agent with the ability to identify the types of events listed below.

6.1.2.1 Packet Sniffer & Unauthorised Protocols

Detects and logs event records when components on the system register with the operating system, enabling the detection of packet sniffers and other non-IP protocols that are not authorised to run on the network.

The Windows operating system allows applications and drivers to register themselves as protocols and receive packets from the network data link driver. This registration is done using a string that identifies the application/protocol.

The Agent detects and logs event records when components on the system register with the operating system for this purpose. This allows the detection of packet sniffers as well as other non-IP protocols that should not be running on the Agent.

6.1.2.2 Port-scans and Ping-scans

Detects port-scans by tracking all TCP "RESET" packets and ICMP "DESTINATION UNREACHABLE" packets that are sent by the system in response to received connection requests. The Agent generates an event record for all responses and sends them to the Management Center.

The Agent detects ping-scans by logging all ICMP "ECHO" requests to the Management Center. The Agent detects port-scans by tracking all TCP "RESET" packets and ICMP "DESTINATION UNREACHABLE" packets that are sent by the system in response to received connection requests. These packets are responses to connection attempts to ports that are not being listened on and are indication that a remote system may be attempting to map the active ports on the system. The Agent generates an event record for all such responses and sends them to the Management Center. If more than five such event records are received from a system within one hour, the port-scan event is stored in the events database; otherwise it is discarded as being a false positive due to random network noise.

Ping-scans are detected in a similar fashion. Every ICMP "ECHO" request is logged to the Management Center. However, unlike the port-scan detection function, ping-scan event records are always discarded unless reported from multiple systems.

6.1.2.3 Syn-flood Attacks

Protects against Syn-flood attacks by tracking real addresses based on prior responses to received packets.

The Agent protects against Syn-flood denial of service attacks by tracking real addresses based on prior responses to received packets. These "good" addresses can always connect, whereas the number of connections from unknown addresses is limited, preventing filling up the operating system host connection table.

6.1.2.4 Malformed Packets

Protects against malformed packet attacks by checking that the received packets conform to the protocol specification and validating various header lengths for consistency.

The Agent protects the system from malformed packets that are received over the network. These packets may be sent by an attacker for various purposes, for example, to exercise a bug in the operating system's network implementation that causes a system crash or to make the system an accomplice in flooding some other victim by responding to broadcast source address. The Agent protects against these types of attacks by checking that received packets conform to the protocol specification and validating various header length fields for consistency. Packets that fail these checks are dropped and an event record is generated.

6.1.2.5 Email Worms

Protects by tracking downloaded email content. If an email message or attachment is downloaded by an email client, read in by a process or itself executed, then attempts to connect to the network or to access the email address book, the Agent displays a prompt to the End User. The End User may allow the operation or terminate the process.

Email worms are malicious scripts or executables that are accidentally or unknowingly downloaded and invoked by an End User and which then sends copies of themselves to other systems/users on the network. The address of the target systems is retrieved from some source on the infected computer such as the End User’s email address book.

The Agent protects against these by tracking downloaded email content. If an email message or attachment is downloaded by an email client, read in by another process, which then attempts to connect to the network or to access the email address book, the Agent displays a prompt to the End User. The End User may choose to allow the operation or terminate the process. If terminated, an event record is sent to the Management Center to be stored in the events database.

6.1.2.6 Keystroke Loggers

Detects applications that attempt to capture system keystrokes.

The Agent provides protection against keystroke loggers by tracking processes which are monitoring keystrokes even when they are not the foreground application. Since this is a legitimate action by some programs such as keyboard macro recorders, the Agent uses additional information such as whether the application accesses the network, the period of time it monitors keystrokes, and file access patterns to distinguish keystroke loggers. Upon detection, the End User is displayed a prompt to select an option to terminate the process or to allow it to monitor the keystrokes. If the user chooses to terminate, the process is terminated and an event record is sent to the Management Center.

6.1.2.7 Code Injection

Detects applications that are marked as downloaded content attempting to write code to space owned by other applications.

The Windows operating system allows applications to register functions to be called when various Windows messages are sent to an application. The application may specify the function to be called for messages sent to any application. This causes the module containing the registered function (call-back function) to be loaded into all processes; this is called code injection.

The Agent intercepts invocations of the registration function to check if the call-back function is located in an executable or shared library that was downloaded from the network. If this occurs, the End User is displayed a prompt to select an option to terminate the process or allow it to continue. If the End User chooses to terminate, the process is terminated and an event record is sent to the Management Center.

6.1.2.8 Trojan Programs (Process Memory Protection)

Detects applications that attempt to interfere with the memory space of other applications and detects Trojans attempting to hide in another executable to escape detection and gain permissions to access other resources.

Certain types of Trojan programs attempt to hide themselves in the memory space of other applications. This is generally accomplished by making use of certain Windows system calls that allow a process to write data and create execution threads in the memory space of other processes.

The Agent intercepts invocations of these system calls and displays a prompt to the End User to select an option to either terminate the process or allow it to continue. If the End User chooses to terminate, the process is terminated and an event record is sent to the Management Center.

6.1.2.9 Password Theft

Detects applications that attempt to steal local system passwords.

The Windows 2000 and NT operating systems store user passwords in an area of the Windows Registry. This area is normally only accessed by the lsass.exe process and is protected through access rights. However, any process running with administrative privileges can modify these access rights and thereby gain access to the passwords. The Agent intercepts all accesses to this area of the Windows Registry and displays a prompt to the End User to select an option to terminate the process or to allow it to continue. If the End User chooses to disallow the action, access is not allowed but the process is not terminated. Either action creates an event record that is sent to the Management Center.

6.1.2.10 Downloaded Executable Files Automatically Invoked Without User Intervention

The Agent protects against downloaded executable files that are automatically invoked without user intervention by tracking files with extensions .exe, .com, and .pif that are downloaded from the network and immediately executed.

Some forms of attacks take place through downloaded executable files that are automatically invoked without user intervention. The Agent protects against this by tracking files of type .exe, .com, and .pif that are downloaded from the network and immediately executed. When a process is created from an executable that is downloaded within 10 seconds of downloading, the Agent displays a prompt to the End User to select an option to terminate the process or allow it to continue. If the End User chooses termination, the process is terminated and an event record is sent to the Management Center.

6.1.2.11 Downloaded Active X Controls

Detects downloaded Active X controls that immediately attempt execution.

Active X controls are a form of executable content that can be downloaded and executed by Web browsers. The Web browser triggers the download by requesting an Active X control to be loaded into the process. If the control is not already present on the system, a download is initiated from a remote server. The Agent intercepts these requests and displays a prompt to the End User to select an option to either allow or deny the download. If allowed, the Active X control is downloaded and allowed to execute. If denied, an error is returned to the process, but the process is allowed to continue, and an event record is sent to the Management Center.

6.1.2.12 Buffer Overflow Exploits

Verifies the return address of the caller of system functions for any process that has accessed the network. If the return address lies in the heap or stack sections of the process address space, the Agent displays a prompt to the End User requesting an action.

Many applications are susceptible to buffer overflow exploits where data that is read from the network overflows some internal buffer and allows the attacker to execute code on the target system. In most cases, the attacker executes code that calls system functions such as creating command shell processes. The Agent verifies the return address of the caller of system functions in any process that has accessed the network. If the return address lies in the heap of stack sections of the processes address space, the Agent displays a prompt to the End User to select an option to terminate the process or allow it to continue. If denied, the process is terminated, and an event record is sent to the Management Center.

6.1.2.13 ICMP Covert Channels

Prevents transmission and reception of unsolicited ICMP response packets.

Because firewalls are often configured to let ICMP packets through, many exploits install Trojan programs that tunnel communications with a remote attacker's system using ICMP packet payloads. To protect against this covert channel, the Agent prevents transmission and reception of unsolicited ICMP response packets, and payloads in ICMP response packets must match the payload in the corresponding ICMP requests.

6.1.3 Program Access Control

Access control rules are the foundation of the security policies configured by the Administrator. The Management Center enables the Administrator to create file access control, network access control, Windows Registry access control, and COM (component object model) access control rules. CSA ships with default rules. Access control decisions are based on security attributes of the subject and object.

Table 1-11 Subject Attributes

Subject Attributes
Executable file name
Executable version

Many of the access control types described below rely on the application run control, which controls whether processes that belong to specific application classes are allowed to run. Applications classes are an abstraction construct used by the CSA policy engine. Application classes are based on the attributes listed above. Because the application class membership of a process can evolve as a result of its execution, a check is made on every resource access attempted by a process.

The Administrator constructs Program Access Control Policy Rules (the security policy) based on attributes such as application class definitions, various resource type definitions, and rules that define allowed or denied interaction between the two. The policy enforced by an Agent is a set of rules that defines the specified behaviour allowed for the processes in each application class. A process may belong to multiple application classes.

The Agent enforces specified actions when a process attempts certain operations such as accessing a file. The Administrator specifies the conditions (application, resource, operation) under which an action is to be taken as part of the security policy. An attempted operation may match multiple rules specifying various sets of conditions, each of which maintains an associated action. The Administrator writes the rules specifying the action as either high priority deny, high priority terminate, allow, query user (default allow), query user (default deny), terminate, deny, and default allow. Rules are default restrictive. No special privileges are granted to a rule without an explicit definition. The table below specifies the priority order of the actions to be taken within each policy.

The priority levels outlined in the following table indicate the manner in which the Agent processes rules. An attempted operation may match multiple rules specifying different sets of conditions, each of which has an associated action. In such a case, the action with the highest priority is taken. In a case where multiple rules are applicable, all of which have the same actions specified, the rules that specify that the action should be logged have priority over the rules that do not. If there are multiple rules that match all criteria, exactly which rule has effect is inconsequential because the effect is the same for all.

Table 1-12 Rule Priority Order

Action	Priority	Description
High priority deny	1	Prevents the attempted action if it matches the specified conditions.
High priority Terminate	2	Prevents the attempted action and terminates the process if it matches the specified conditions.
Allow	3	Allows the attempted operation if it matches the specified conditions.

Query user (default allow)	4	If the operation matches the specified conditions, a prompt is displayed to the user asking if the attempted operation should be allowed. If the user does not respond, the operation is allowed.
Query user (default deny)	5	If the operation matches the specified conditions, a prompt is displayed to the user asking if the attempted operation should be allowed. If the user does not respond, the operation is denied.
Terminate	6	Prevents the attempted action and terminates the process if it matches the specified conditions
Deny	7	Prevents the attempted action if it matches the specified conditions.
Default allow	8	If the operation does not match any of the conditions (rules) in the policy for the Agent, the operation is allowed by default.

6.1.3.1 File Access Control

Controls access to file resources on fixed disks and network storage. The Agent may take one or more defined actions in response to an attempted access to a file. File access control rules allow or deny what operations (read, write) a selected application is permitted to perform on specific files. File access control is based on the application attempting to access the file, and the operation (read, write) attempting to take action on the file. CSA ships with default rules, which will be used for the evaluation.

Table 1-13 File Attributes

File Attributes
Path
Operation requested

6.1.3.2 Network Access Control

Controls access to and from network resources for TCP and UDP based communication. Network access control rules control access to specified network services. Access control to network resources is determined by the action that is allowed or denied, the application attempting to access the service or address, the direction (client, server) of the communication, the service a system is attempting to use, and the address a system is attempting to communicate with. CSA ships with default rules, which will be used for the evaluation.

Table 1-14 Network Attributes

Network Attributes
Target address
Operation requested
Direction (client, server)
Service (port)

6.1.3.3 Windows Registry Access Control

Controls write access to the Windows Registry. The Windows Registry access control rules allow or deny selected applications from writing to specified Windows Registry keys. The Windows Registry access control is based on the action that is allowed or denied, and the application attempting to write to the Windows Registry keys and values. CSA ships with default rules, which will be used for the evaluation.

Table 1-15 Windows Registry Attributes

Windows Registry Attributes
Windows Registry key
Operation requested

6.1.3.4 COM Component Access Control

Controls access to the Windows Component Object Model (COM) components installed on a system. COM components expose numerous interfaces that may be exploited by malicious code. COM component access control rules allow or deny selected applications from accessing specified COM components. COM component access control rules are determined by the action that is allowed or denied and the application accessing the COM component. CSA ships with default rules, which will be used for the evaluation.

Table 1-16 COM Attributes

COM Attributes
Name of COM component
Operation requested

6.1.4 Management Functions

The Management Center provides the Administrator with encrypted, remote access to a Web-based GUI that is used to manage the TOE. The Web-based GUI is served by the Apache v. 1.3.20 Web server. The management GUI requires the Administrator to login. Detailed information on the Administrator login is given in the Identification & Authentication section above.

In addition to the administrative functions that are detailed in their respective sections above, the Administrator can manage the Program Access Control Policy Rules. Management includes the ability to view, edit, enable, or disable any of the Program Access Control Policy Rules. These rules are used by the Agents to enforce programmatic access to files, the network, the Windows Registry, and COM components. Collectively the Program Access Control Policy Rules form the security policy for the Agents. The security policy is published to the registered Agents through a secure link provided by the Web server.

The Administrator can create multiple policies specific to certain hosts or groups of hosts that have Agents. These policies consider attributes of the subjects and specify one of six different actions if the rule matches. A detailed listing of these attributes and actions and how they can be correlated is given in the Program Access Control section above.

It is important to note, the Administrator is the user of the system who can log into the Management Center, thereby being the only user who can create, modify, or delete security policies.

6.2 Security Assurance Measures and Rationale

The assurance measures provided by the TOE satisfy all of the assurance requirements listed in the following table, which provides a reference between each TOE assurance requirement and the related vendor documentation that satisfies each requirement.

Table 1-17 Assurance Measures and Rationale

Assurance Component	Documentation Satisfying Component	Rationale
ACM_CAP.2	Common Criteria CM v1.0	This document describes the CM system complete with access controls in place during code and documentation development. It also describes the naming and versioning conventions used by the CM system.
ADO_DEL.1	Delivery and Operation v1.0	This describes the secure delivery procedures for the TOE.
ADO_IGS.1	Installing Management Center for Cisco Security Agents 4.5.1.1	This describes the installation procedures for both the Management Center and the Cisco Security Agent in the evaluated configuration. The secure management of the TOE is also described.

ADV_FSP.1	The three documents making up the FSP are: CC Admin table-K, FSP Screen shots for Table-e, and Function Mapping –F.	These describe the functions, the interfaces and the functional behavior at these interfaces. Errors and exception handling is also described.
ADV_HLD.1	CSA High Level Design EAL2 V0-1	This document describes the security functions in terms of subsystems and indicates the entry points into the TOE and the interactions between the subsystems.
ADV_RCR.1	RCR V0.1	This document shows the correspondence between the High Level Design and the FSP and between the FSP and the TSS. All SFRs are mapped.
AGD_ADM.1	Using Management Center for Cisco Security Agents 4.5.1.1	This describes the Administrator's role and the secure management of the TOE in the evaluated configuration
AGD_USR.1	Using Management Center for Cisco Security Agents 4.5.1.1	This describes the User's role and the secure usage of the TOE in the evaluated configuration
ATE_COV.1	CSA Coverage Analysis for Common Criteria Certification v0.1	This document maps the testing effort to the functions and hence SFRs that were tested.
ATE_FUN.1		
ATE_IND.2	Done by the lab	This independently tests a subset of the vendor tests as supplied and independently tests the CSA as required. The lab will provide test documentation for this work unit.
AVA_SOF.1	Developer Strength of Function Analysis Document (AVA_SOF.1)	This document explains the overall strength of function claim for the TOE (SOF-Basic) and indicates that there are no probabilistic or permutational functions requiring an SOF claim in the ST.
AVA_VLA.1	Developer Vulnerability Analysis v0.1 Lab vulnerability testing.	This document describes the vulnerabilities the TOE may be susceptible to and the TOE's mitigation for these vulnerabilities. The lab conducts vulnerability testing based on and building on the developer's vulnerability assessment.

6.3 Rationale for TOE Security Functions

The following table shows the mapping between the Security Functional Requirements and the CSA security functions enforced by the TOE, which are listed above.

Table 1-18 Mapping of Security Functional Requirements to TOE Security Functions

	Audit	Monitoring and Detection	Program Access Control	Management Functions
FAU_ARP.1		**X**		
FAU_GEN.1-*NIAP-0347*	**X**		**X**	
FAU_SAA.3		**X**		
FAU_SAR.1	**X**			
FAU_SAR.3	**X**			
FAU_SEL.1-*NIAP-0407*	**X**			
FAU_STG.1-*NIAP-0422*	**X**			
FDP_ACC.1			**X**	
FDP_ACF.1-*NIAP-0407*			**X**	
FMT_MOF.1			**X**	**X**
FMT_MSA.1			**X**	
FMT_MSA.3			**X**	
FMT_MTD.1 (1)				**X**
FMT_MTD.1 (2)	**X**			
FMT_SMF.1	**X**			**X**
FMT_SMR.1				**X**
FPT_RVM.1		**X**		

The following section provides a rationale supporting how each CSA security function satisfies each Security Functional Requirement.

FAU_ARP.1 Security Alarms - The Monitoring and Detection security function satisfies FAU_ARP.1 by notifying the End User of many types of network and malicious code attacks.

FAU_GEN.1-***NIAP-0347***Audit Data Generation - The Audit security function satisfies FAU_GEN.1-***NIAP-0410*** by recording all events that are recorded in the events database. The event record documents the system events according to designated time frames, event severity levels, and the system that generated the event, thus supporting the requirements for FAU_GEN.1-***NIAP-0347***, Audit Data Generation.

FAU_SAA.3 Simple Attack Heuristics - The Monitoring and Detection security function satisfies FAU_SAA.3 by detecting many types of network and malicious code attacks.

FAU_SAR.1 Audit Review - The Audit security function satisfies FAU_SAR.1 by providing the ability to view the audit records via the

Management Center. The Management Center provides various formats for viewing the audit records. The reports are displayed in either plain HTML or using an ActiveX viewer.

FAU_SAR.3 Selectable Audit Review - The Audit security function satisfies FAU_SAR.3 by providing the ability to search, sort, and order the event logs based on designated time frames, event severity levels, and the system that generated the event.

FAU_SEL.1-***NIAP-0407*** Selective Audit - The Audit security function satisfies FAU_SEL.1-***NIAP-0407*** by providing the Management Center with the ability to only audit records based on object identity, subject identity, host identity, event type, time interval, and severity level.

FAU_STG.1-***NIAP-0422*** Protected Audit Trail Storage - The Audit security function satisfies FAU_STG.1-***NIAP-0422*** by providing measures to protect the stored audit records by limiting write access to the events database to the Administrator.

FDP_ACC.1 Subset Access Control - The Program Access Control security function satisfies FDP_ACC.1 by providing the Administrator the ability to create file access control, network access control, Windows Registry access control and COM component access control rules.

FDP_ACF.1-***NIAP-0407*** Security Attribute Based Access Control - The Program Access Control security function satisfies FDP_ACF.1-***NIAP-0407*** by providing the Administrator the ability to create file access control, network access control, Windows Registry access control and COM component access control rules.

FMT_MOF.1 Management of Security Functions Behaviour - The Program Access Control security function satisfies FMT_MOF.1 by providing the ability to enforce access control rules as defined by the Administrator. Also, the Management Functions security function provides the Administrator's ability to view, edit, enable, or disable any of the Program Access Control Policy Rules.

FMT_MSA.1 Management of Security Attributes - The Program Access Control security function satisfies FMT_MSA.1 by providing the ability to enforce the file access control, network access control, Windows Registry access control and COM component access control rules as defined by the Administrator.

FMT_MSA.3 Static Attribute Initialization - The Program Access Control security function satisfies FMT_MSA.3 by providing default restrictive values for the file access control, network access control, Windows Registry access control and COM component access control rules.

FMT_MTD.1 (1) Management of TSF Data (1)- The Management Functions security function satisfies FMT_MTD.1 (1) by providing the Administrator the ability to create, modify, or delete the Program Access Control Policy Rules.

FMT_MTD.1 (2) Management of TSF Data (2)- The Audit security function satisfies FMT_MTD.1 (2) by providing the Administrator the ability to read, and delete audit data.

FMT_SMF.1 Specification of Management Functions - FMT_SMF.1 is satisfied by the Audit security function, which controls audit records, and the Management Functions security function, which controls Program Access Control Policy Rules.

FMT_SMR.1 Security Roles - The Management Functions security function satisfies FMT_SMR.1 by establishing the Administrator as the only role supported by the TOE.

FPT_RVM.1 Non-Bypassability of the TSP - The Monitoring & Detection security function satisfies FPT_RVM.1 by providing that only authorized accesses to the TSF and TSF data is permitted.

6.4 Appropriate Strength of Function Claim

No security functions include a password mechanism that is probabilistic or permutational. The rationale for choosing SOF-basic is based on the low attack potential of threats identified in this ST.

CHAPTER 7

7 Protection Profile Claims

This chapter provides detailed information in reference to the Protection Profile conformance identification that appears in Chapter 1, Section 1.4 Protection Profile Conformance.

7.1 Protection Profile Reference

This Security Target does not claim conformance to any registered Protection Profile.

7.2 Protection Profile Refinements

This Security Target does not claim conformance to any registered Protection Profile.

7.3 Protection Profile Additions

This Security Target does not claim conformance to any registered Protection Profile.

7.4 Protection Profile Rationale

This Security Target does not claim conformance to any registered Protection Profile.

CHAPTER 8

8 Rationale

This Security Target does not claim conformance to any Protection Profiles.

8.1 Security Objectives Rationale

Sections 4.3 – 4.5.1 provide the security objectives rationale.

8.2 Security Requirements Rationale

Sections 5.6 Rationale for TOE Security Functional Requirements – 5.9 provide the security requirements rationale.

8.3 TOE Summary Specification Rationale

Sections 6.2 Security Assurance Measures and Rationale - 6.4 Appropriate Strength of Function Claim provide the TSS rationale.

8.4 Protection Profile Claims Rationale

This Security Target does not claim conformance to any Protection Profiles