Shavlik Technologies, LLC
Shavlik Security Suite v8.0

Security Target

Evaluation Assurance Level: EAL3+
Document Version: 1.0

Prepared for:

Shavlik Technologies, LLC
2665 Long Lake Road, Suite 400
Roseville, MN 55113
Phone: (800) 690-6911

http://www.shavlik.com

Prepared by:

Corsec Security, Inc.
10340 Democracy Lane, Suite 201
Fairfax, VA 22030
Phone: (703) 267-6050

http://www.corsec.com

Table of Contents

Table of Figures

Table of Tables

1 Security Target Introduction

This section identifies the Security Target (ST), Target of Evaluation (TOE), and the ST organization. The Target of Evaluation is the Shavlik Security Suite, and will hereafter be referred to as the TOE throughout this document. The TOE is an automated patch and configuration management solution.

1.1 Purpose

This ST contains the following sections to provide mapping of the Security Environment to the Security Requirements that the TOE meets in order to remove, diminish or mitigate the defined threats:

- Security Target Introduction (Section 1) – Provides a brief summary of the ST contents and describes the organization of other sections within this document. It also provides an overview of the TOE security functions and describes the physical and logical scope for the TOE, as well as the ST and TOE references.
- Conformance Claims (Section 2) – Provides the identification of any Common Criteria (CC), ST Protection Profile, and Evaluation Assurance Level (EAL) package claims. It also identifies whether the ST contains extended security requirements.
- Security Problem Definition (Section 3) – Describes the threats, organizational security policies, and assumptions that pertain to the TOE and its environment.
- Security Objectives (Section 4) – Identifies the security objectives that are satisfied by the TOE and its environment.
- Extended Components Definition (Section 5) – Identifies new components (extended Security Functional Requirements (SFRs) and extended Security Assurance Requirements (SARs)) that are not included in CC Part 2 or CC Part 3.
- Security Requirements (Section 6) – Presents the SFRs and SARs met by the TOE.
- TOE Summary Specification (Section 7) – Describes the security functions provided by the TOE that satisfy the security functional requirements and objectives.
- Rationale (Section 8) - Presents the rationale for the security objectives, requirements, and SFR dependencies as to their consistency, completeness, and suitability.
- Acronyms (Section 9) – Defines the acronyms used within this ST.

1.2 Security Target and TOE References

Table 1 – ST and TOE References

ST Title	Shavlik Technologies, LLC Shavlik Security Suite v8.0 Security Target
ST Version	Version 1.0
ST Author	Corsec Security, Inc. Nathan Lee, Greg Milliken
ST Publication Date	June 16, 2010
TOE Reference	Shavlik Security Suite v8.0

1.3 TOE Overview

The TOE Overview summarizes the usage and major security features of the TOE. The TOE Overview provides a context for the TOE evaluation by identifying the TOE type, describing the product, and defining the specific evaluated configuration.

The TOE is the Shavlik Security Suite software. The Shavlik Security Suite simplifies and automates critical Information Technology (IT) operations, enabling organizations to reduce their spending – less time, less money, less IT staff – on necessary functions including system discovery, patch management, and configuration management. Shavlik optimizes IT tasks, freeing up IT staff for initiatives that grow your business. The Suite allows organizations to:

- Manage security patches
- Assess and update system security configurations
- Review all system security information from one easy-to-use dashboard

Shavlik Security Suite bundles Shavlik's industry-leading management products in one package. The Suite provides full and current versions of:

- **Shavlik NetChk® Protect v7.5 build 2716:** Simplifies enterprise-wide vulnerability management, providing agent-less patch scanning, and patch deployment (via the NetChk Scheduler component, described below) from one console. NetChk Protect also includes an optional NetChk Agent that can perform scans of the system on which it is installed, allowing the NetChk Agent to act in a stand-alone capacity (without the need to communicate with a centralized NetChk Protect server).
- **Shavlik NetChk® Configure v4.2 build 20:** A powerful agent-less compliance management solution that simplifies and automates the management of critical system and security configurations. It enables organizations to conform to emerging regulations, meet compliance objectives, lower costs, and reduce the organization's risk of exposure.
- **Shavlik Security Intelligence™ v4.2 build 02032010:** An intuitive, customizable Web-based dashboard that allows organizations to integrate multiple data sources (including Shavlik NetChk Protect, Shavlik NetChk Configure, and others) at one location, providing one unified and comprehensive view. Although Shavlik Security Intelligence (SSI) is sold with the TOE and is included within the TOE boundary, none of the claimed security functionality is enforced by SSI. SSI does not enforce or support any of the SFRs claimed in this ST.

These three TOE software components can be deployed in a variety of configurations, the most common of which is depicted in Figure 1 below. The software runs on Microsoft Windows operating systems (OS) (XP, Vista, Server 2003, and Server 2008) and general-purpose computing hardware platforms that are not included in the TOE.

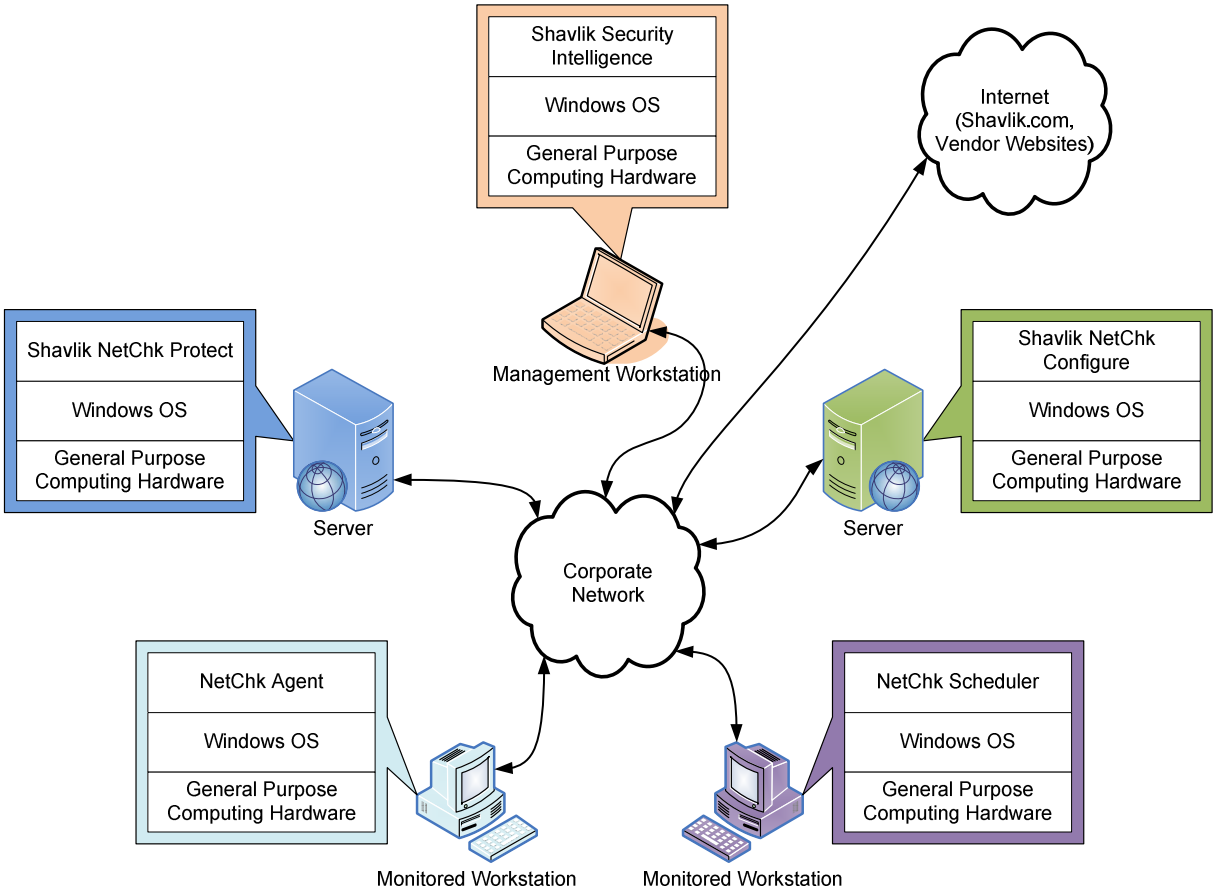

Figure 1 – Deployment Configuration of the TOE

1.3.1 Brief Description of the Components of the TOE

The TOE components shown in Figure 1 above are briefly described in the following subsections.

1.3.1.1 Shavlik NetChk Protect v7.5 build 2716

NetChk Protect allows network administrators to schedule automatic scans of the Windows-based machines connected to their networks for needed patches, and to respond by automatically deploying patches when vulnerabilities are found. Unlike competing products, NetChk Protect implements an *agent-less* client/server architecture, whereby the NetChk Protect server component scans the workstations or servers on the configured network, determines what patches are needed, and deploys the patches as a bundle (created from a local patch distribution server) to the workstations or servers for scheduled installation by the NetChk Scheduler.

1.3.1.2 NetChk Scheduler v7.5 build 2716 (Part of NetChk Protect v7.5)

NetChk Scheduler is an application that NetChk Protect copies to managed workstations or servers (if it is not already present) and installs as a running service on those machines as part of a patch deployment. NetChk Scheduler then installs the patch bundle copied to the workstation or server by NetChk Protect, and optionally removes itself from the workstation or server when patch deployment is complete.

1.3.1.3 NetChk Agent v7.5 build 2716 (Part of NetChk Protect v7.5)

NetChk Agent is a "stand-alone" alternative to the client/server agent-less architecture provided by NetChk Protect and NetChk Scheduler. NetChk Agent runs directly on a managed workstation or server, providing the same functionality of NetChk Protect and NetChk Scheduler, but NetChk Agent only scans the machine on which it is installed.

1.3.1.4 Shavlik NetChk Configure v4.2 build 20

NetChk Configure allows network administrators to schedule automatic scans of the configurations of the servers and workstations connected to their networks and to automatically analyze these configurations for conformance to various administrator-defined policies. Policy violations are identified for action. Like NetChk Protect, NetChk Configure implements an *agent-less* client/server architecture, whereby NetChk Configure scans the Windows-based computers on the configured network.

1.3.1.5 Shavlik Security Intelligence v4.2 build 02032010

SSI is an intuitive, customizable Web-based dashboard that makes critical security information easily accessible, giving administrators the power to simply measure risk and policy compliance across an organizations' network.

1.3.2 TOE Environment

The evaluated deployment configuration of the TOE requires the following environmental components in order to function properly:

- Server running a supported version of Microsoft Windows[1] (for Shavlik NetChk Protect)
- Server running a supported version of Microsoft Windows[1] (for Shavlik NetChk Configure)
- Server running a supported version of Microsoft Windows (for Shavlik Security Intelligence)
- Managed server or workstation running a supported version of Microsoft Windows (for NetChk Scheduler)
- Managed server or workstation running a supported version of Microsoft Windows (for NetChk Agent)
- The Microsoft Windows OS that the TOE is installed on
- Network switch (with connection to the Internet)

1.4 TOE Description

This section primarily addresses the physical and logical components of the TOE included in the evaluation.

1.4.1 Physical Scope

The TOE is comprised of both software components and guidance documentation. Figure 2 below illustrates the physical scope and the physical boundary of the overall solution and ties together all of the components of the TOE and the constituents of the TOE Environment. The guidance documentation included in the TOE is as follows:

- Shavlik NetChk Protect 7.5 Administration Guide
- Shavlik NetChk Protect 7.5 Installation & Setup Guide

[1] The FIPS 140-2 validated cryptographic module is included with Windows.

- Shavlik NetChk Configure 4.2 Installation Guide
- Shavlik NetChk Configure 4.2 Administration Guide
- Shavlik Security Suite 8.0 Common Criteria Guidance Supplement

The software-only TOE is a patch and configuration management software suite which is installed on general-purpose computing hardware running Microsoft Windows operating systems (OS). The TOE is installed on a network in a distributed manner as depicted in the figure below. The TOE boundary includes the Shavlik Security Suite software but excludes the underlying operating system and hardware platform.

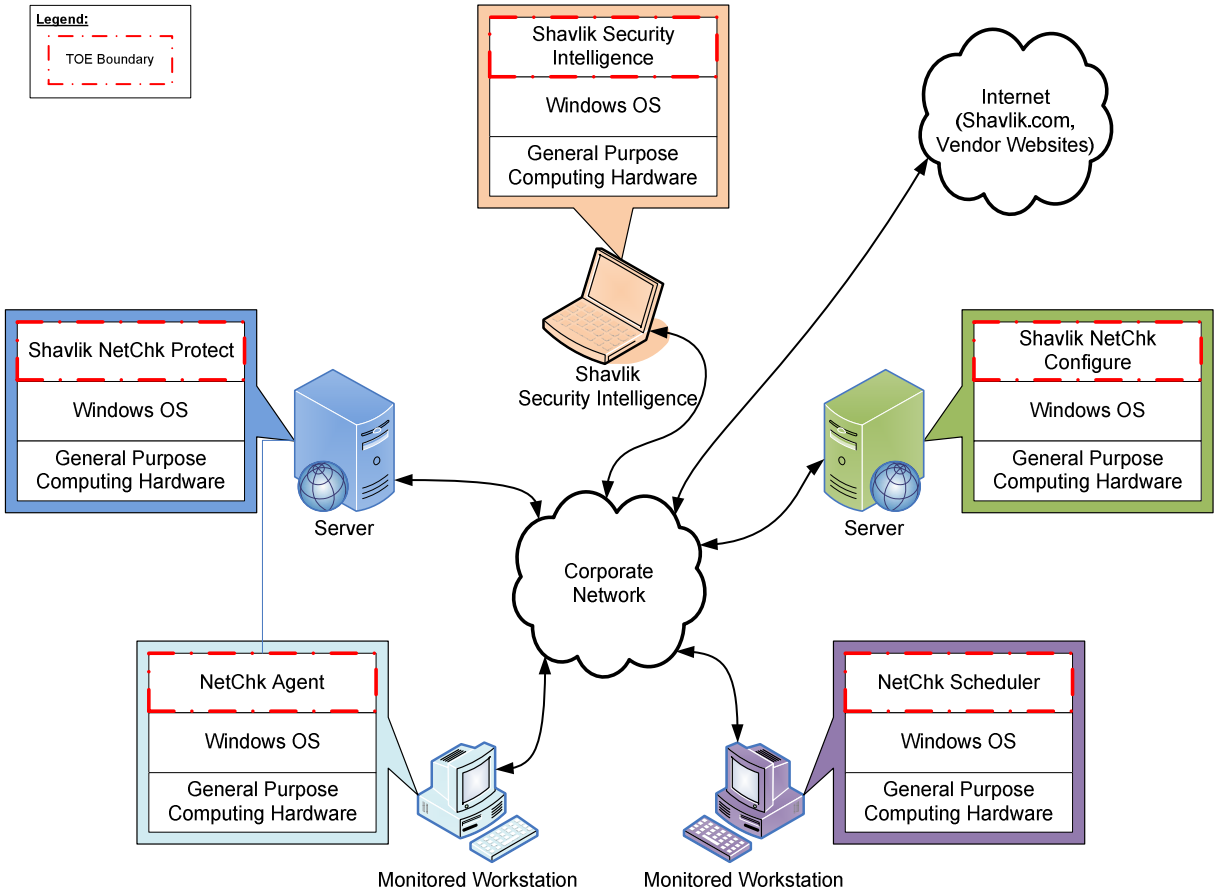

Figure 2 – Physical TOE Boundary

1.4.2 Logical Scope

The security functional requirements implemented by the TOE are usefully grouped under the following Security Function Classes:

- Security Audit
- User Data Protection

- Identification and Authentication
- Security Management
- Protection of the TSF[2]
- Resource Utilization
- Data Collection

1.4.2.1 Security Audit

The TOE generates audit records each time a machine is scanned, a patch is applied, and a security violation is discovered, and allows authorized administrators to review the audit records.

1.4.2.2 User Data Protection

The TOE implements an access control security functional policy (SFP) (Access Control SFP) which is concerned with mediating access to NetChk Protect and NetChk Configure administrative functions; an information flow control SFP concerned with mediating access to machine-scanning functionality and patch-deployment functionality (Protect SFP); and an information flow control SFP concerned with mediating access to machine-scanning functionality and configuration-deployment functionality (Configure SFP).

1.4.2.3 Identification and Authentication

The TOE maintains the unique Windows account identifier and assigns a role for each user for access control and auditing purposes.

1.4.2.4 Security Management

The TOE provides three security management functions, upon which access control is enforced:

- Management of security functions behavior
- Management of security attributes
- Management of TSF data

1.4.2.5 Protection of the TSF

Shavlik controlled patch and configuration data is protected from modification while being transmitted between separate parts of the TOE. Shavlik controlled patch and configuration data will only be used if the integrity of the data is determined to be valid. The integrity of TOE software is also verified upon execution of a TOE component and will only allow itself to execute or be executed by properly verified software. Integrity checking is based on digital signatures attached to Shavlik's data and TOE executable code. The cryptographic functionality related to creating and verifying digital signatures takes place in the Windows operating system in a FIPS 140-2 validated cryptographic module. The Windows operating system is outside of the TOE boundary and part of the TOE Environment.

Cryptography is provided by the cryptographic modules listed in Appendix A.

1.4.2.6 Resource Utilization

The TOE limits the number of machines that can be scanned simultaneously.

[2] TSF: TOE Security Function

1.4.2.7 Data Collection

When a scan is run, the TOE generates, stores, protects, and analyzes collection logs for potential action by an administrator.

1.4.3 Product Functionality Not Included in the TOE

The following product functionality is not part of the evaluated configuration of the TOE:

- Shavlik NetChk Protect “ping-back mode”
- Malware detection and removal
- Application control

2 Conformance Claims

This section provides the identification for any CC, Protection Profile (PP), and EAL package conformance claims. Rationale is provided for any extensions or augmentations to the conformance claims. Rationale for CC and PP conformance claims can be found in Section 8.1.

Table 2 – CC and PP Conformance

Common Criteria (CC) Identification and Conformance	Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 2, September 2007; CC Part 2 extended; CC Part 3 conformant.
PP Identification	None
Evaluation Assurance Level	EAL3 augmented with flaw remediation (ALC_FLR.2)

3 Security Problem Definition

This section describes the security aspects of the environment in which the TOE will be used and the manner in which the TOE is expected to be employed. It provides the statement of the TOE security environment, which identifies and explains all:

- Known and presumed threats countered by either the TOE or by the security environment
- Organizational security policies with which the TOE must comply
- Assumptions about the secure usage of the TOE, including physical, personnel and connectivity aspects

3.1 Threats to Security

This section identifies the threats to the IT assets against which protection is required by the TOE or by the security environment. The threat agents are divided into two categories:

- Attackers who are not TOE users: They have public knowledge of how the TOE operates and are assumed to possess a low skill level, limited resources to alter TOE configuration settings/parameters and no physical access to the TOE.
- TOE users: They have extensive knowledge of how the TOE operates and are assumed to possess a high skill level, moderate resources to alter TOE configuration settings/parameters and physical access to the TOE. (TOE users are, however, assumed not to be willfully hostile to the TOE.)

Both are assumed to have a low level of motivation. The IT assets requiring protection are the user data saved on or transitioning through the TOE and the hosts on the protected network. Removal, diminution and mitigation of the threats are through the objectives identified in Section 4 - Security Objectives.

The following threats are applicable:

Table 3 – Threats

Name	Description
T.AUDACC	Persons may not be accountable for the actions that they conduct because the audit records cannot be reviewed, thus allowing an attacker to escape detection.
T.MASQUERADE	An attacker may masquerade as another entity in order to gain unauthorized access to data or TOE resources.
T.TSF_COMP	An attacker or user may cause through an unsophisticated attack, the TSF to be inappropriately accessed (viewed, modified, or deleted).
T.UNAUTH	A user or administrator may gain access to security data on the TOE, even though the user is not authorized in accordance with the TOE security policy.
T.MODIFY	An attacker may attempt to modify or replace TSF data as it is being transmitted between physically separate parts of the TOE.
T.INT_ATK	An attacker may exploit internal weaknesses in the TOE implementation to gain access to data without authorization.
T.BADSTATE	An attacker may exploit vulnerabilities in monitored IT entities that reach an insecure state without the network administrators becoming aware.

3.2 Organizational Security Policies

There are no Organizational Security Policies defined for this Security Target.

3.3 Assumptions

This section describes the security aspects of the intended environment for the evaluated TOE. The operational environment must be managed in accordance with assurance requirement documentation for delivery, operation, and user guidance. The following specific conditions are required to ensure the security of the TOE and are assumed to exist in an environment where this TOE is employed.

Table 4 – Assumptions

Name	Description
A.INSTALL	It is assumed that the TOE is installed on the appropriate, dedicated hardware and operating system.
A.NETCON	It is assumed that the TOE environment provides the network connectivity required to allow the TOE to provide secure patch and configuration management functions.
A.TIMESTAMP	It is assumed that the IT environment provides the TOE with the necessary reliable timestamps.
A.LOCATE	It is assumed that the TOE is located within a controlled access facility.
A.MANAGE	It is assumed that there are one or more competent individuals assigned to manage the TOE and the security of the information it contains.
A.NOEVIL	It is assumed that the users who manage the TOE are not careless, negligent, or willfully hostile, and follow all guidance.
A.FIREWALL	It is assumed that all ports needed for proper operation of the TOE will be opened at the firewall. Also, any firewall settings necessary for the TOE's operation will be configured to allow the TOE to operate.
A.OS_AUTH	It is assumed that the TOE environment will provide identification and authentication functions for users attempting to manage and use the TOE.
A.SECCOMM	It is assumed that the environment provides a sufficient level of protection to secure communications between distribution servers (if deployed), agents (if deployed) and other TOE components.
A.FIPS	A FIPS 140-2 validated cryptographic module in the TOE Environment must provide all cryptographic functionality for the TOE.

4 Security Objectives

Security objectives are concise, abstract statements of the intended solution to the problem defined by the security problem definition (see Section 3). The set of security objectives for a TOE form a high-level solution to the security problem. This high-level solution is divided into two part-wise solutions: the security objectives for the TOE, and the security objectives for the TOE's operational environment. This section identifies the security objectives for the TOE and its supporting environment.

4.1 Security Objectives for the TOE

The specific security objectives for the TOE are as follows:

Table 5 – Security Objectives for the TOE

Name	Description
O.LOG	The TOE must record events of security relevance and provide authorized administrators with the ability to review the recorded events.
O.MANAGE	The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.
O.ROLE	The TOE must be able to associate users and administrators with the appropriate role after the user or administrator authenticates.
O.INTEGRITY	The TOE must protect data being transmitted to physically separate parts of the TOE from unauthorized modification.
O.INT_ATK	The TOE implementation must be able to mitigate attacks to stored executable code and thread overuse.
O.MONITOR	The TOE must be able to monitor machines on the network to ensure that they exist in a secure state and alert TOE users if a system enters an insecure state.

4.2 Security Objectives for the Operational Environment

4.2.1 IT Security Objectives

The following IT security objectives are to be satisfied by the environment:

Table 6 – IT Security Objectives

Name	Description
OE.TIME	The operating system where the TOE is installed must provide reliable timestamps to the TOE.
OE.OS_AUTH	The operating system where the TOE is installed must provide authentication and identification of individuals attempting to use the TOE.

OE.PLATFORM	The TOE environment must include hardware and an operating system for the TOE to be installed on.
OE.FIREWALL	The firewall must have all ports needed for proper operations of the TOE opened.
OE.SECCOMM	The TOE environment must provide mechanisms to secure communications between TOE agents, distribution servers, and other TOE components.
OE.CONNECT	The TOE environment must be implemented such that the TOE is appropriately located within and connected to the network to perform its intended function.
OE.FIPS	The operating system that the TOE is installed upon must provide a FIPS 140-2 validated cryptographic module for the TOE to use to perform cryptographic functions.

4.2.2 Non-IT Security Objectives

The following non-IT environment security objectives are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software. Thus, they will be satisfied largely through application of procedural or administrative measures.

Table 7 – Non-IT Security Objectives

Name	Description
OE.PHYCAL	Those responsible for the TOE must ensure that the TOE is protected from any physical attack.
OE.MANAGE	Sites deploying the TOE will provide administrators for the TOE who are not careless, negligent, or willfully hostile, are appropriately trained and follow all administrator guidance. TOE administrators will ensure the system is used securely, including management of the audit trail.
OE.REVIEW	The configuration of the TOE will be inspected on a regular basis to ensure that the configuration continues to meet the organization's security policies in the face of: • Changes to the TOE configuration • Changes in the security objectives • Changes in the threats presented by the hostile network • Changes (additions and deletions) in the services available between the hostile network and the corporate network

5 Extended Components Definition

This section defines the extended SFRs met by the TOE. These requirements are presented following the conventions identified in Section 6.1.1.

5.1 Extended TOE Security Functional Components

This section specifies the extended SFRs for the TOE. The extended SFRs are organized by class. Table 8 identifies all extended SFRs implemented by the TOE

Table 8 – Extended TOE Security Functional Requirements

Name	Description
FDC_ANA.1 (EXP)	System Analysis
FDC_SCN.1 (EXP)	System Scan
FDC_STG.1 (EXP)	Scanned Data Storage
FAU_GEN.1 (EXP)	Audit data generation

5.1.1 Class FDC: Data Collection and Analysis

Data Collection and Analysis functions involve:

- Scanning systems to obtain data,
- Storing the collected data,
- Performing analysis on collected data and presenting analytical results to administrators in a format that allows administrators to take appropriate actions.

The FDC: Data Collection and Analysis class was modeled after the CC FAU: Security audit class. The extended family and related components for FDC_ANA: System Analysis were modeled after the CC family and related components for FAU_SAA: Security audit analysis. The extended family FDC_SCN: System Scan was modeled after the CC family FAU_GEN: Security audit data generation. The extended family FDC_STG: Scanned Data Storage was modeled after the CC family FAU_STG: Security audit event storage.

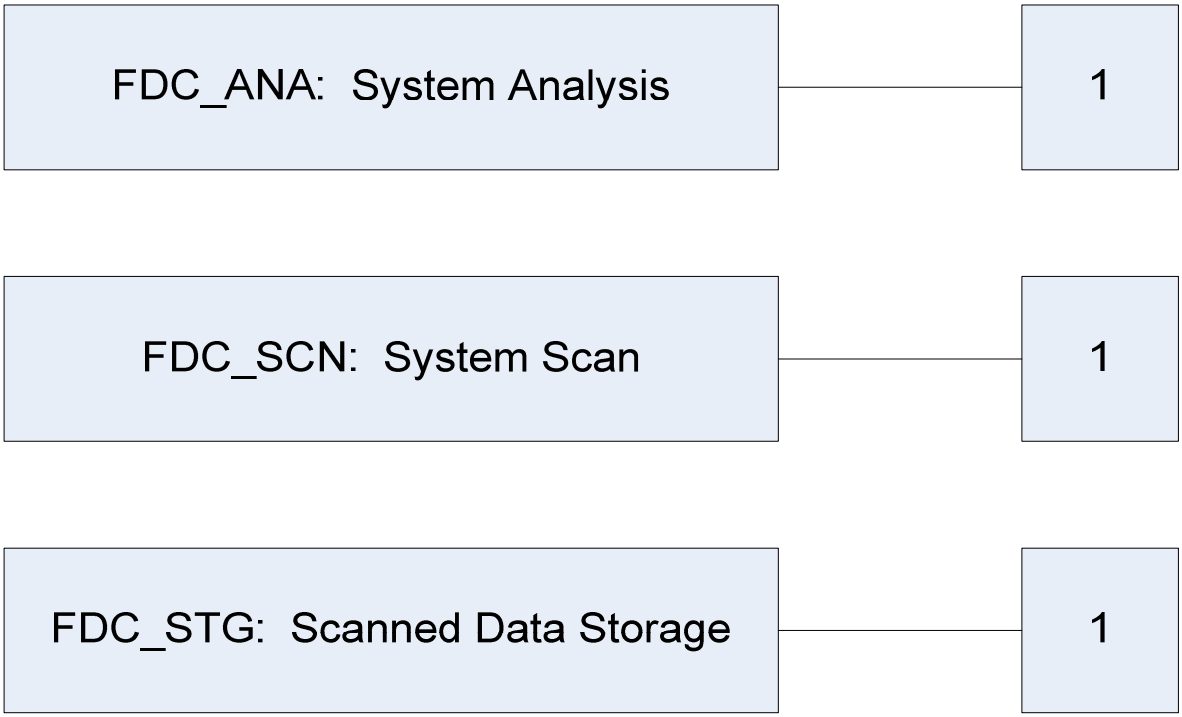

Figure 3 – FDC: Data Collection and Analysis Class Decomposition

5.1.1.1 FDC_ANA: System Analysis

Family Behaviour

This family defines the requirements for the use of tools for the analysis of collected data and that allow administrators to react to potential security violations found during analysis of collected data.

Component Leveling

Figure 4 – FDC_ANA: System Analysis family decomposition

FDC_ANA.1: System Analysis provides the capability to analyze collected data and present the results to administrators in a way that easily allows the administrators to respond to potential security violations found during the analysis.

Management: FDC_ANA.1 (EXP)

The following actions could be considered for the management functions in FMT:

- Maintenance (deletion, modification, addition) of the analysis rules or the set of systems the rules are applied to.

Audit: FDC_ANA.1 (EXP)

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

- Minimal: Identity of the entity who initiated a scan or deployed a patch.
- Minimal: Identity of the scanned machines, list of security violations discovered, list of configuration changes made, and list of patches applied to machines.

FDC_ANA.1 (EXP) System Analysis

Hierarchical to: No other components

Dependencies: FDC_SCN.1 System Scan (EXP)

This component provides the capability to analyze collected data and present the results to administrators in a way that easily allows the administrators to respond to potential security violations found during the analysis.

FDC_ANA.1.1 (EXP)

The TSF shall be able to apply a set of rules in monitoring the scanned data and based upon these rules indicate potential security violations:

a) compare applied patches against a list of potential patches and indicate which applications do not have all patches applied;

b) compare a machines current configuration against a baseline configuration and indicate which configuration settings do not match the baseline configuration.

FDC_ANA.1.2 (EXP)

The TSF shall enforce the following set of rules for monitoring scanned data:

a) [assignment: *Information Flow Control Policy to be applied to scanned data*];

b) [assignment: *any other rules*].

FDC_ANA.1.3 (EXP)

The TSF shall be able to indicate a possible security violation to [assignment: *list of users with permission to review analytical results*] and allow [assignment: *list of users with permission to apply patches or configuration updates to scanned machines*] to address security violations that are discovered.

5.1.1.2 FDC_SCN: System Scan

Family Behaviour

This family defines the requirements for scanning systems to retrieve data about their patch deployment and configuration state.

Component Leveling

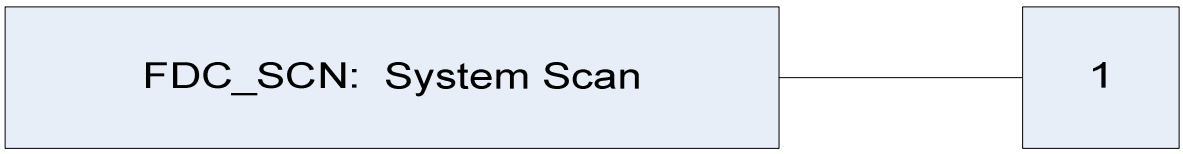

Figure 5 – FDC_SCN: System Scan family decomposition

FDC_SCN.1: System Scan defines the scanning function and specifies which machines will have a scan performed on them.

Management: FDC_SCN.1 (EXP)

- There are no management activities foreseen.

Audit: FDC_SCN.1 (EXP)

- There are no auditable events foreseen.

FDC_SCN.1 (EXP) System Scan

Hierarchical to: No other components

Dependencies: None.

This component provides the ability to scan targeted machines for data related to patch levels and security configurations.

FDC_SCN.1.1 (EXP)

The System shall be able to collect the following information from the targeted IT System resource(s):

a) patch levels for [assignment: *list of applications to monitor patch levels for*];

b) system configuration parameters for the [assignment: *list of configuration policies*]; and

c) no other information.

FDC_SCN.1.2 (EXP)

The TSF shall record within each scan file at least the following information:

1. Date and time of the scan, list of machines scanned, identity of the entity who initiated the scan, list of security violations discovered during the scan; and

2. no other information.

5.1.1.3 FDC_STG: Scanned Data Storage

Family Behaviour

This family defines the requirements for protecting stored scan data.

Component Leveling

Figure 6 – FDC_STG: Scanned Data Storage family decomposition

FDC_STG.1: Scanned Data Storage, defines how the TSF protects stored scan data from unauthorized modification or deletion.

Management: FDC_STG.1 (EXP)

- There are no management activities foreseen.

Audit: FDC_STG.1 (EXP)

- There are no auditable events foreseen.

FDC_STG.1 (EXP) Scanned Data Storage

Hierarchical to: No other components

Dependencies: FDC_SCN.1 System Scan (EXP)

This component provides the ability to protect stored scan data from unauthorized deletion and modification.

FDC_STG.1.1 (EXP)

The TSF shall protect the stored scan data from unauthorized deletion.

FDC_STG.1.2 (EXP)

The TSF shall be able to prevent unauthorized modifications to the stored scan data.

5.1.2 Class FAU: Security Audit

Explicitly stated Security Audit functions involve:

- Generation of audit data for the TOE.

The FAU: Security Audit class was modeled after the CC FAU: Security Audit class. The extended family and related components for FAU_GEN: Audit data generation were modeled after the CC family and related components for FAU_GEN: Security audit data generation.

Figure 7 – FAU: Security Audit Class Decomposition

5.1.2.1 FAU_GEN: Security audit data generation

Family Behaviour

This family defines requirements for recording the occurrence of security relevant events that take place under TSF control. This family identifies the level of auditing, enumerates the types of events that shall be auditable by the TSF, and identifies the minimum set of audit-related information that should be provided within various audit record types.

Component Leveling

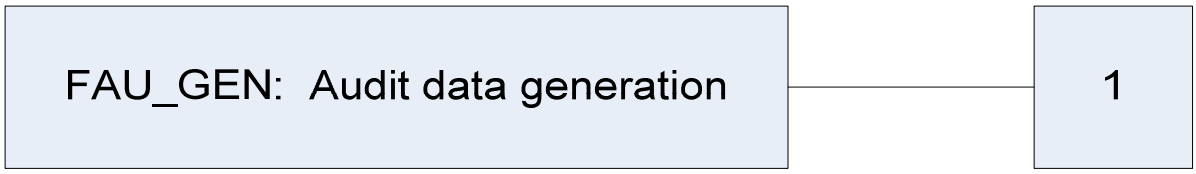

Figure 8 – FAU_GEN: Audit data generation family decomposition

FAU_GEN.1: Audit data generation provides the capability to generate audit records for security-relevant events.

Management: FAU_GEN.1 (EXP)

The following actions could be considered for the management functions in FMT:

There are no management activities foreseen.

Audit: FAU_GEN.1 (EXP)

There are no auditable events foreseen.

FAU_GEN.1 (EXP) Audit data generation

Hierarchical to: No other components

Dependencies: FPT_STM.1 Reliable time stamps

This component provides the capability to generate audit records for security-relevant events and enumerates the events to be audited.

FAU_GEN.1.1 (EXP)

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

a) All auditable events for the [selection, choose one of: *minimum, basic, detailed, not specified*] level of audit; and

b) [assignment: *other specifically defined auditable events*].

FAU_GEN.1.2 (EXP)

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

a) Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [assignment: *other audit relevant information*].

5.2 Extended TOE Security Assurance Components

There are no extended SARs defined for this Security Target.

6 Security Requirements

This section defines the SFRs and SARs met by the TOE. These requirements are presented following the conventions identified in Section 6.1.1.

6.1.1 Conventions

There are several font variations used within this ST. Selected presentation choices are discussed here to aid the Security Target reader.

The CC allows for assignment, refinement, selection and iteration operations to be performed on security functional requirements. All of these operations are used within this ST. These operations are performed as described in Parts 2 and 3 of the CC, and are shown as follows:

- Completed assignment statements are identified using [*italicized text within brackets*].
- Completed selection statements are identified using [*underlined italicized text within brackets*].
- Refinements are identified using **bold text**. Any text removed is stricken (Example: ~~TSF Data~~) and should be considered as a refinement.
- Extended Functional and Assurance Requirements are identified using "(EXP)" at the end of the short name.
- Iterations are identified by appending a letter following the component title. For example, FAU_GEN.1a Audit Data Generation would be the first iteration and FAU_GEN.1b Audit Data Generation would be the second iteration.

6.2 Security Functional Requirements

This section specifies the SFRs for the TOE. This section organizes the SFRs by CC class. Table 9 identifies all SFRs implemented by the TOE and indicates the ST operations performed on each requirement.

Table 9 – TOE Security Functional Requirements

Name	Description	S	A	R	I
FAU_GEN.1 (EXP)	Audit data generation	✓	✓		
FAU_SAR.1	Audit review		✓		
FDP_ACC.1	Subset access control		✓		
FDP_ACF.1	Security attribute based access control		✓		
FDP_IFC.1a	Subset information flow control (Protect)		✓		✓
FDP_IFF.1a	Simple security attributes (Protect)		✓		✓
FDP_IFC.1b	Subset information flow control (Configure)		✓		✓
FDP_IFF.1b	Simple security attributes (Configure)		✓		✓
FIA_ATD.1	User attribute definition		✓		
FMT_MOF.1	Management of security functions behaviour	✓	✓		

FMT_MSA.1a	Management of security attributes (user roles)	✓	✓		✓
FMT_MSA.1b	Management of security attributes (machine properties)	✓	✓		✓
FMT_MSA.3a	Static attribute initialisation (Access Control SFP)	✓	✓		✓
FMT_MSA.3b	Static attribute initialisation (Protect SFP)	✓	✓		✓
FMT_MSA.3c	Static attribute initialization (Configure SFP)	✓	✓		✓
FMT_MTD.1	Management of TSF data	✓	✓		
FMT_SMF.1	Specification of management functions		✓		
FMT_SMR.1	Security roles		✓		
FPT_ITT.1	Basic internal TSF data transfer protection	✓			
FPT_ITT.3	TSF data integrity monitoring	✓	✓		
FPT_TST.1	TSF testing	✓	✓	✓	
FRU_RSA.1	Maximum quotas	✓			
FDC_ANA.1 (EXP)	System Analysis		✓		
FDC_SCN.1 (EXP)	System Scan		✓		
FDC_STG.1 (EXP)	Scanned Data Storage				

Note: S=Selection; A=Assignment; R=Refinement; I=Iteration

6.2.1 Class FAU: Security Audit

FAU_GEN.1 (EXP) Audit data generation

Hierarchical to: No other components.

FAU_GEN.1.1

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

- All auditable events, for the [*not specified*] level of audit; and
- [*list of machines scanned, list of patches applied, list of discovered security violations*].

FAU_GEN.1.2

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

- Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
- For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*no other information*].

Dependencies: FPT_STM.1 Reliable time stamps

FAU_SAR.1 Audit review

Hierarchical to: No other components.

FAU_SAR.1.1

The TSF shall provide [*NetChk Configure and NetChk Protect administrators*] with the capability to read [*all audit data*] 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.

Dependencies: FAU_GEN.1 Audit data generation

6.2.2 Class FDP: User Data Protection

FDP_ACC.1 Subset access control

Hierarchical to: No other components.

FDP_ACC.1.1

The TSF shall enforce the [*Access Control SFP*] on [

Subjects: Administrators attempting to establish an interactive session with the TOE

Objects: User interface menu items, policies, machine groups, scans, product features

Operations: All interactions between the subjects and objects identified above

].

Dependencies: FDP_ACF.1 Security attribute based access control

FDP_ACF.1 Security attribute based access control

Hierarchical to: No other components.

FDP_ACF.1.1

The TSF shall enforce the [*Access Control SFP*] to objects based on the following: [

Subject attributes:

1. Role

2. Windows user identifier (ID)

and Object attributes:

1. Permissions assigned to objects

2. Absence of permissions assigned to objects

].

FDP_ACF.1.2

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

1. *If a NetChk Configure administrator requests access to an object then access is granted.*

2. If a NetChk Protect administrator requests access to an object and the administrator's role has permission to access that object then access is granted.

3. If none of the above rules apply, access is denied.

].

FDP_ACF.1.3

The TSF shall explicitly authorise access of subjects to objects based on the following additional rules: [*no other rules*].

FDP_ACF.1.4

The TSF shall explicitly deny access of subjects to objects based on the [*no other rules*].

Dependencies: **FDP_ACC.1 Subset access control**
FMT_MSA.3 Static attribute initialization

FDP_IFC.1a Subset information flow control (Protect)

Hierarchical to: No other components.

FDP_IFC.1.1a

The TSF shall enforce the [*Protect SFP*] on [

Subjects: Machines that are members of machine groups

Information: data obtained by scanning the machines and patches to be applied to machines

Operations: Analysis of scanned data against a patch list, application of patches to machines

].

Dependencies: **FDP_IFF.1 Simple security attributes**

FDP_IFC.1b Subset information flow control (Configure)

Hierarchical to: No other components.

FDP_IFC.1.1b

The TSF shall enforce the [*Configure SFP*] on [

Subjects: Machines that are members of machine groups

Information: data obtained by scanning the machines and configuration updates to be applied to the machines

Operations: Analysis of scan data against an administrator-defined rule set, application of configuration updates to machines

].

Dependencies: **FDP_IFF.1 Simple security attributes**

FDP_IFF.1a Simple security attributes (Protect)

Hierarchical to: No other components.

FDP_IFF.1.1a

The TSF shall enforce the [*Protect SFP*] based on the following types of subject and information security attributes: [

Subject Attributes:

1. *Machine group membership*

Information Attributes:

1. *Machine of origin*
2. *Installed applications*
3. *Installed patches*
4. *Digital signature of the patch file (if applicable)*

].

FDP_IFF.1.2a

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

a) *An authorized administrator requests that a machine be scanned or*

b) *An authorized administrator requests that a patch be applied to a machine*

].

FDP_IFF.1.3a

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

FDP_IFF.1.4a

The TSF shall explicitly authorise an information flow based on the following rules: [*an authorized administrator with appropriate permissions has scheduled a scan to be performed at some point in the future*].

FDP_IFF.1.5a

The TSF shall explicitly deny an information flow based on the following rules: [*the patch does not match its signature (if applicable)*].

Dependencies: **FDP_IFC.1 Subset information flow control**
FMT_MSA.3 Static attribute initialization

FDP_IFF.1b Simple security attributes (Configure)

Hierarchical to: No other components.

FDP_IFF.1.1b

The TSF shall enforce the [*Configure SFP*] based on the following types of subject and information security attributes: [

Subject Attributes:

1. *Machine group membership*

Information Attributes:

1. *Machine of origin*
2. *Registry values*
3. *Services*
4. *User Rights*
5. *File Access Control Lists*
6. *Directory Access Control Lists*

].

FDP_IFF.1.2b

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

a) *An authorized administrator requests that a machine be scanned or*

b) *An authorized administrator requests that a configuration update be applied to a machine*

].

FDP_IFF.1.3b

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

FDP_IFF.1.4b

The TSF shall explicitly authorise an information flow based on the following rules: [*an authorized administrator with appropriate permissions has scheduled a scan to be performed at some point in the future*].

FDP_IFF.1.5b

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

Dependencies: FDP_IFC.1 Subset information flow control
FMT_MSA.3 Static attribute initialization

6.2.3 Class FIA: Identification and Authentication

FIA_ATD.1 User attribute definition

Hierarchical to: No other components.

FIA_ATD.1.1

The TSF shall maintain the following list of security attributes belonging to individual users: [*Role, Windows account identifier*].

Dependencies: No dependencies

6.2.4 Class FMT: Security Management

FMT_MOF.1 Management of security functions behaviour

Hierarchical to: No other components.

FMT_MOF.1.1

The TSF shall restrict the ability to [*determine the behaviour of, modify the behaviour of*] the functions [*the list of functions in the 'Permissions' column of Table 10*] to [*the roles indicated in the 'Role' column of Table 10*].

Table 10 – Security functions behaviour by role

Component	Role	Permissions
NetChk Configure	NetChk Configure Administrator	• Create, delete, modify machine groups • Create, delete, modify policies • Create, delete, modify checks • Initiate, schedule scans • Initiate, schedule configuration updates • Create, view reports • Manage scan data in the NetChk Configure database
NetChk Protect	NetChk Protect Administrator	• Create, delete, modify users • Create, delete, modify machine groups • Initiate, schedule scans • Initiate, schedule patch updates • Create, delete, modify patch groups • Create, view reports • Create, delete, modify deployment templates • Delete scan/deployment results • Create, delete, modify agent policy • Install, remove NetChk Agent
	Full User	• Create, delete, modify machine groups • Initiate, schedule scans • Initiate, schedule patch updates • Create, delete, modify patch groups • Create, view reports • Create, delete, modify deployment templates • Delete scan/deployment results • Create, delete, modify agent policy • Install, remove NetChk Agent
	Scan and Report Only	• Initiate, schedule scans • Create, view reports
	Deploy and Report Only	• Initiate, schedule patch updates • Create, view reports
	Report Only	• Create, view reports

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MSA.1a Management of security attributes (User roles)

Hierarchical to: No other components.

FMT_MSA.1.1a

The TSF shall enforce the [*Access Control SFP*] to restrict the ability to [*change_default, modify*] the security attributes [*role*] to [*NetChk Protect Administrator*].

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MSA.1b Management of security attributes (Machine properties)

Hierarchical to: No other components.

FMT_MSA.1.1b

The TSF shall enforce the [*Protect SFP and Configure SFP*] to restrict the ability to [*change_default, query, modify, delete*] the security attributes [*machine group membership*] to [*NetChk Configure Administrators, NetChk Protect Administrators, Full Users*].

Dependencies: [FDP_ACC.1 Subset access control or
FDP_IFC.1 Subset information flow control]
FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_MSA.3a Static attribute initialization (Access Control SFP)

Hierarchical to: No other components.

FMT_MSA.3.1a

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

FMT_MSA.3.2a

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

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3b Static attribute initialization (Protect SFP)

Hierarchical to: No other components.

FMT_MSA.3.1b

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

FMT_MSA.3.2b

The TSF shall allow the [*NetChk Protect Administrator, Full User, Deploy and Report Only*] to specify alternative initial values to override the default values when an object or information is created.

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MSA.3c Static attribute initialization (Configure SFP)

Hierarchical to: No other components.

FMT_MSA.3.1c

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

FMT_MSA.3.2c

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

Dependencies: FMT_MSA.1 Management of security attributes
FMT_SMR.1 Security roles

FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

FMT_MTD.1.1

The TSF shall restrict the ability to [*query, delete*] the [*data from scanned machines*] to [*the NetChk Protect Administrator, NetChk Protect Full User, and NetChk Configure Administrator*].

Dependencies: FMT_SMF.1 Specification of management functions
FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

Hierarchical to: No other components.

FMT_SMF.1.1

The TSF shall be capable of performing the following management functions: [*management of security functions behaviour, management of security attributes, management of TSF data*].

Dependencies: No Dependencies

FMT_SMR.1 Security roles

Hierarchical to: No other components.

FMT_SMR.1.1

The TSF shall maintain the roles [

For the NetChk Configure application:

1. *NetChk Configure Administrator*

For the NetChk Protect application:

1. *NetChk Protect Administrator*
2. *Full User*
3. *Scan and Report Only*
4. *Deploy and Report Only*
5. *Report Only*

].

FMT_SMR.1.2

The TSF shall be able to associate users with roles.

Dependencies: FIA_UID.1 Timing of identification

6.2.5 Class FPT: Protection of the TSF

FPT_ITT.1 Basic internal TSF data transfer protection

Hierarchical to: No other components.

FPT_ITT.1.1

The TSF shall protect TSF data from [*modification*] when it is transmitted between separate parts of the TOE.

Dependencies: No dependencies

FPT_ITT.3 TSF data integrity monitoring

Hierarchical to: No other components.

FPT_ITT.3.1

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

FPT_ITT.3.2

Upon detection of a data integrity error, the TSF shall take the following actions: [*drop the corrupted data*].

Dependencies: FPT_ITT.1 Basic internal TSF data transfer protection

FPT_TST.1 TSF testing

Hierarchical to: No other components.

FPT_TST.1.1

The TSF shall run a suite of self tests [*at the conditions [during execution of a TOE component]*] to demonstrate the correct operation of [*the TSF*].

FPT_TST.1.2

The TSF shall ~~provide authorised users with the capability to~~ **automatically** verify the integrity of [*digitally signed TSF data*].

FPT_TST.1.3

The TSF shall ~~provide authorised users with the capability to~~ **automatically** verify the integrity of stored TSF executable code.

Dependencies: No dependencies

6.2.6 Class FRU: Resource Utilization

FRU_RSA.1 Maximum quotas

Hierarchical to: No other components.

FRU_RSA.1.1

The TSF shall enforce maximum quotas of the following resources: [*threads dedicated to scanning machines*] that [*a defined group of users*] can use [*simultaneously*].

Dependencies: No dependencies

6.2.7 Class FDC: Data Collection and Analysis (EXP)

FDC_ANA.1 System Analysis (EXP)

Hierarchical to: No other components

FDC_ANA.1.1 (EXP)

The TSF shall be able to apply a set of rules in monitoring the scanned data and based upon these rules indicate potential security violations.

a) compare applied patches against a list of potential patches and indicate which applications do not have all patches applied;

b) compare a machines current configuration against a baseline configuration and indicate which configuration settings do not match the baseline configuration.

FDC_ANA.1.2 (EXP)

The TSF shall enforce the following set of rules for monitoring scanned data:

a) [*Protect SFP, Configure SFP*];

b) [*no other rules*].

FDC_ANA.1.3 (EXP)

The TSF shall be able to indicate a possible security violation to [*NetChk Configure administrators, NetChk Protect Administrators, Full Users, Scan and Report Only, and Deploy and Report Only*] and allow [*NetChk Configure administrators, NetChk Protect Administrators, Full User, and Deploy and Report Only*] to address security violations that are discovered.

Dependencies: FDC_SCN.1 System Scan (EXP).

FDC_SCN.1 System Scan (EXP)

Hierarchical to: No other components

FDC_SCN.1.1 (EXP)

The System shall be able to collect the following information from the targeted IT System resource(s):

a) patch levels for [*the list of applications supported under the Protect SFP*];

b) system configuration parameters for the [*policies and checks supported under the Configure SFP*]; and

c) no other information.

FDC_SCN.1.2 (EXP)

The TSF shall record within each scan file at least the following information:

a) Date and time of the scan, list of machines scanned, identity of the entity who initiated the scan, list of security violations discovered during the scan; and

b) no other information.

Dependencies: **None.**

FDC_STG.1 Scanned Data Storage (EXP)

Hierarchical to: **No other components**

FDC_STG.1.1 (EXP)

The TSF shall protect the stored scan data from unauthorized deletion.

FDC_STG.1.2 (EXP)

The TSF shall be able to prevent unauthorized modifications to the stored scan data.

Dependencies: **FDC_SCN.1 System Scan (EXP).**

6.3 Security Assurance Requirements

This section defines the assurance requirements for the TOE. Assurance requirements are taken from the CC Part 3 and are EAL3 augmented with ALC_FLR.2. Table 11 – Assurance Requirements summarizes the requirements.

Table 11 – Assurance Requirements

Assurance Requirements	
Class ALC : Life Cycle Support	ALC_CMC.3 Authorisation controls
	ALC_CMS.3 Implementation representation CM[3] coverage
	ALC_DEL.1 Delivery Procedures
	ALC_DVS.1 Identification of security measures
	ALC_LCD.1 Developer defined life-cycle model
	ALC_FLR.2 Flaw reporting procedures
Class ADV: Development	ADV_ARC.1 Security Architecture Description
	ADV_FSP.3 Functional specification with complete summary
	ADV_TDS.2 Architectural design
Class AGD: Guidance documents	AGD_OPE.1 Operational user guidance
	AGD_PRE.1 Preparative procedures
Class ATE: Tests	ATE_COV.2 Analysis of coverage
	ATE_DPT.1 Testing: basic design
	ATE_FUN.1 Functional testing
	ATE_IND.2 Independent testing – sample
Class AVA: Vulnerability assessment	AVA_VAN.2 Vulnerability analysis

[3] CM – Configuration Management

7 TOE Summary Specification

This section presents information to detail how the TOE meets the functional requirements described in previous sections of this ST.

7.1 TOE Security Functions

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

Table 12 – Mapping of TOE Security Functions to Security Functional Requirements

TOE Security Function	SFR ID	Description
Security Audit	FAU_GEN.1 (EXP)	Audit data generation
	FAU_SAR.1	Audit review
User Data Protection	FDP_ACC.1	Subset access control
	FDP_ACF.1	Security attribute based access control
	FDP_IFC.1a	Subset information flow control (Protect)
	FDP_IFF.1a	Simple security attributes (Protect)
	FDP_IFC.1b	Subset information flow control (Configure)
	FDP_IFF.1b	Simple security attributes (Configure)
Identification and Authentication	FIA_ATD.1	User attribute definition
Security Management	FMT_MOF.1	Management of security functions behaviour
	FMT_MSA.1a	Management of security attributes (user roles)
	FMT_MSA.1b	Management of security attributes (machine properties)
	FMT_MSA.3a	Static attribute initialisation (Access Control SFP)
	FMT_MSA.3b	Static attribute initialisation (Protect SFP)
	FMT_MSA.3c	Static attribute initialization (Configure SFP)

	FMT_MTD.1	Management of TSF data
	FMT_SMF.1	Specification of management functions
	FMT_SMR.1	Security roles
Protection of TOE Security Functions	FPT_ITT.1	Basic internal TSF data transfer protection
	FPT_ITT.3	TSF data integrity monitoring
	FPT_TST.1	TSF testing
Resource Utilization	FRU_RSA.1	Maximum quotas
Data Collection and Analysis	FDC_ANA.1 (EXP)	System Analysis
	FDC_SCN.1 (EXP)	System Scan
	FDC_STG.1 (EXP)	Scanned Data Storage

7.1.1 Security Audit

The TOE generates audit logs that contain the following information:

- Date and time of the event
- Type of event
- Subject identity (if applicable)
- Outcome (success or failure) of the event

The TOE generates audit records each time a machine is scanned, a patch is applied, and a security violation is discovered.

TOE Security Functional Requirements Satisfied: FAU_GEN.1 (EXP).

The TOE provides audit logs for administrators to review in a form suitable for the administrators to interpret the information in the logs. The logs are available via the NetChk Protect or NetChk Configure server applications. Only authorized administrators are permitted to view the audit records.

TOE Security Functional Requirements Satisfied: FAU_SAR.1.

7.1.2 User Data Protection

The TOE implements one access control SFP and two information flow control SFPs, which are described below.

7.1.2.1 Access Control SFP

The Access Control SFP is concerned with mediating access to NetChk Protect and NetChk Configure administrative functions. When a user (a "subject") invokes the NetChk Protect console application, the console application checks the user's assigned role and then only grants permission to access the management options ("objects") for which that user's role is authorized. When a user invokes the NetChk Configure console application, the console application grants the user permission to access all management options. See Section 7.1.4 below for more details.

TOE Security Functional Requirements Satisfied: FDP_ACC.1, FDP_ACF.1.

7.1.2.2 Protect SFP

The Protect SFP is concerned with mediating access to machine-scanning functionality and patch-deployment functionality. Only authorized administrators may initiate a manual (immediate) or scheduled (delayed) machine scan or patch deployment. The integrity of a patch update file is verified before it is used, and any patch update file that fails integrity verification is not used. Integrity verification is based on digital signatures of the patch data. The digital signatures are created and verified by a FIPS 140-2 validated cryptographic module on the Windows operating system.

TOE Security Functional Requirements Satisfied: FDP_IFC.1(a), FDP_IFF.1(a).

7.1.2.3 Configure SFP

The Configure SFP is concerned with mediating access to machine-scanning functionality and configuration-deployment functionality. Only authorized administrators may initiate a manual (immediate) or scheduled (delayed) machine scan or configuration deployment. The integrity of a configuration update file is verified before it is used, and any configuration update file that fails integrity verification is not used. Integrity verification is based on digital signatures of the configuration data. The digital signatures are created and verified by a FIPS 140-2 validated cryptographic module on the Windows operating system.

TOE Security Functional Requirements Satisfied: FDP_IFC.1(b), FDP_IFF.1(b).

7.1.3 Identification and Authentication

The users of the TOE are authenticated by the underlying Windows operating system before the TOE is invoked. After the TOE is invoked, it uses the user's Windows account identifier (his Windows username) and his role (assigned by the TOE) for identification and access control purposes.

TOE Security Functional Requirements Satisfied: FIA_ATD.1.

7.1.4 Security Management

The TOE provides three security management functions:

- Management of security functions behavior
- Management of security attributes
- Management of TSF data

TOE Security Functional Requirements Satisfied: FMT_SMF.1.

The TOE implements administrative roles and associates each TOE user with one or more of these roles. The NetChk Configure application implements one administrative role ("NetChk Configure Administrator"), and the NetChk Protect application implements five administrative roles:

- Administrator
- Full User
- Scan and Report Only
- Deploy and Report Only
- Report Only

TOE Security Functional Requirements Satisfied: FMT_SMR.1.

Administrative roles are used by the TOE to determine which users may manage the behavior of the TOE's security functions. NetChk Configure implements a basic access control mechanism: only authenticated users (*i.e.* users that were authenticated by the underlying OS) may manage the security functions, and every authenticated user has full management authority within NetChk Configure.

NetChk Protect implements a more robust access control mechanism: only OS-authenticated users may manage the security functions, and the TOE determines which NetChk Protect security functions each administrator may manage based on his assigned role and the permissions available to that role. The table in FMT_MOF.1 above provides this access control matrix.

TOE Security Functional Requirements Satisfied: FMT_MOF.1.

Administrative roles are also used by the TOE to determine which users may manage user roles (for NetChk Protect) and machine group membership (for both NetChk Protect and NetChk Configure). FMT_MSA.1(a) and FMT_MSA.1(b) provide these access control matrices.

TOE Security Functional Requirements Satisfied: FMT_MSA.1(a), FMT_MSA.1(b).

The TOE manages the Access Control SFP, the Protect SFP, and the Configure SFP to provide restrictive default values for SFP security attributes. These attributes can be overridden by users with authorized roles. The attribute override permission matrices for these SFRs are provided in FMT_MSA.3(a), FMT_MSA.3(b), and FMT_MSA.3(c) above.

TOE Security Functional Requirements Satisfied: FMT_MSA.3(a), FMT_MSA.3(b), FMT_MSA.3(c).

The TOE protects access to patch data, vulnerability data, and configuration data, only allowing authorized administrators to view, modify, or delete the data. FMT_MTD.1 above provides the access control matrix for these datasets.

TOE Security Functional Requirements Satisfied: FMT_MTD.1.

7.1.5 Protection of the TSF

Shavlik digitally signs all Shavlik patch and configuration data pushed to a machine for deployment. The integrity of the data is verified on the target machine prior to installation, and if the patch fails integrity verification, the TOE does not install it. Integrity verification is based on digital signatures of the patch data. The digital signatures are verified by a FIPS 140-2 validated cryptographic module on the Windows operating system.

TOE Security Functional Requirements Satisfied: FPT_ITT.1, FPT_ITT.3.

In order to prevent tampering by malicious software (such as viruses), each critical executable file and library file composing the TOE is digitally signed by Shavlik. The TOE verifies the integrity of stored signed code prior to allowing a Shavlik executable or library to run another Shavlik binary file. Integrity verification is based on digital signatures of the stored executable code. The digital signatures are created and verified by a FIPS 140-2 validated cryptographic module on the Windows operating system.

TOE Security Functional Requirements Satisfied: FPT_TST.1.

7.1.6 Resource Utilization

In order to prevent resource exhaustion, the TOE limits the number of simultaneous scans that administrators may initiate. By default, NetChk Protect and NetChk Configure will each allow up to 64 simultaneous scans; NetChk Protect can be configured to allow up to 256 simultaneous scans.

TOE Security Functional Requirements Satisfied: FRU_RSA.1.

7.1.7 Data Collection and Analysis

When a scan is run, the TOE generates collection logs that contain the following information:

- Date and time of the scan
- List of machines scanned
- Identity of the entity (user or process on behalf of a user) who initiated the scan
- List of installed and missing patches (for NetChk Protect)
- System configuration parameters (for NetChk Configure)
- List of security violations discovered during the scan (for NetChk Configure)

TOE Security Functional Requirements Satisfied: FDC_SCN.1 (EXP).

The TOE protects the scan data collection logs from unauthorized deletion and modification. Only authorized administrators may clear the logs or delete scan data.

TOE Security Functional Requirements Satisfied: FDC_STG.1 (EXP).

After scan data is collected, the TOE performs automated analysis of the scan data to identify missing patches or incorrect or noncompliant configurations. When potential security violations (missing patches or noncompliant configurations) are detected, the Protect SFP and Configure SFP are enforced when allowing a user to view and address the violations. The access control matrix specifying which administrators may view and address violations is specified in FDC_ANA.1 (EXP) above.

TOE Security Functional Requirements Satisfied: FDC_ANA.1 (EXP).

8 Rationale

8.1 Conformance Claims Rationale

This Security Target extends Part 2 and conforms to part 3 of the Common Criteria Standard for Information Technology Security Evaluations, version 3.1 revision 2. Extended requirements from the FDC class are based on SFRs from the Security Audit (FAU) class.

8.2 Security Objectives Rationale

This section provides a rationale for the existence of each threat, policy statement, and assumption that composes the Security Target. Sections 8.2.1, 8.2.2, and 8.2.3 demonstrate the mappings between the threats, policies, and assumptions to the security objectives is complete. The following discussion provides detailed evidence of coverage for each threat, policy, and assumption.

8.2.1 Security Objectives Rationale Relating to Threats

Table 13 – Threats:Objectives Mapping

Threats	Objectives	Rationale
T.AUDACC Persons may not be accountable for the actions that they conduct because the audit records cannot be reviewed, thus allowing an attacker to escape detection.	O.LOG The TOE must record events of security relevance and provide authorized administrators with the ability to review the recorded events.	O.LOG counters this threat by ensuring that an audit trail of management events on the TOE is preserved.
	OE.TIME The operating system where the TOE is installed must provide reliable timestamps to the TOE.	OE.TIME counters this threat by ensuring that accurate timestamps are provided for all audit records, allowing the order of events to be preserved.
T.MASQUERADE An attacker may masquerade as another entity in order to gain unauthorized access to data or TOE resources.	OE.OS_AUTH The operating system where the TOE is installed must provide authentication and identification of individuals attempting to use the TOE.	OE.OS_AUTH counters this threat by ensuring that the operating system identifies and authenticates TOE users.
	O.ROLE The TOE must be able to associate users and administrators with the appropriate role after the user or administrator authenticates.	O.ROLE counters this threat by ensuring that the TOE is able to associate users with roles according to their operating system user identifier.
T.TSF_COMP An attacker or user may cause through an unsophisticated attack, the TSF to be inappropriately accessed (viewed, modified, or deleted).	O.MANAGE The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.MANAGE counters this threat by restricting the management functions of the TOE to authorized users.

T.UNAUTH A user or administrator may gain access to security data on the TOE, even though the user is not authorized in accordance with the TOE security policy.	O.MANAGE The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	O.MANAGE counters this threat by restricting the management functions of the TOE to authorized users.
	OE.OS_AUTH The operating system where the TOE is installed must provide authentication and identification of individuals attempting to use the TOE.	OE.OS_AUTH counters this threat by ensuring that the operating system identifies and authenticates all TOE users.
	O.ROLE The TOE must be able to associate users and administrators with the appropriate role after the user or administrator authenticates.	O.ROLE counters this threat by ensuring that users are associated with roles while logged into the TOE.
T.MODIFY An attacker may attempt to modify or replace TSF data as it is being transmitted between physically separate parts of the TOE.	O.INTEGRITY The TOE must protect data being transmitted to physically separate parts of the TOE from unauthorized modification.	O.INTEGRITY counters this threat by ensuring that data transferred between physically separate parts of the TOE is not modified or replaced during transmission.
T.INT_ATK An attacker may exploit internal weaknesses in the TOE implementation to gain access to data without authorization.	O.INT_ATK The TOE implementation must be able to mitigate attacks to stored executable code and thread overuse.	O.INT_ATK counters this threat by ensuring that the TOE is implemented in such a way as to prevent attackers from substituting TOE executable code and preventing the overuse of threads.
T.BADSTATE An attacker may exploit vulnerabilities in monitored IT entities that reach an insecure state without the network administrators becoming aware.	O.MONITOR The TOE must be able to monitor machines on the network to ensure that they exist in a secure state and alert TOE users if a system enters an insecure state.	O.MONITOR counters this threat by ensuring that systems on the network are monitored by the TOE and that the TOE alerts TOE users when a security violation occurs.

Every Threat is mapped to one or more Objectives in the table above. This complete mapping demonstrates that the defined security objectives counter all defined threats.

8.2.2 Security Objectives Rationale Relating to Policies

There are no policies defined for this Security Target.

8.2.3 Security Objectives Rationale Relating to Assumptions

Table 14 – Assumptions:Objectives Mapping

Assumptions	Objectives	Rationale
A.INSTALL It is assumed that the TOE is installed on the appropriate, dedicated hardware and operating system.	OE.MANAGE Sites deploying the TOE will provide administrators for the TOE who are not careless, negligent, or willfully hostile, are appropriately trained and follow all administrator guidance. TOE administrators will ensure the system is used securely, including management of the audit trail.	OE.MANAGE upholds this assumption by ensuring that the TOE administrators read and follow the guidance for installation and deployment of the TOE.
	OE.PLATFORM The TOE environment must include hardware and an operating system for the TOE to be installed on.	OE.PLATFORM upholds this assumption by ensuring that an appropriate operating system and hardware is available for the TOE to be installed on.
A.NETCON It is assumed that the TOE environment provides the network connectivity required to allow the TOE to provide secure patch and configuration management functions.	OE.CONNECT The TOE environment must be implemented such that the TOE is appropriately located within and connected to the network to perform its intended function.	OE.CONNECT upholds this assumption by ensuring that the environment provides the TOE with the appropriate configuration to provide secure patch and configuration management functions.
A.TIMESTAMP It is assumed that the IT environment provides the TOE with the necessary reliable timestamps.	OE.TIME The operating system where the TOE is installed must provide reliable timestamps to the TOE.	OE.TIME upholds this assumption by ensuring that the operating system where the TOE is installed will provide reliable time stamps for the TOE.
A.LOCATE It is assumed that the TOE is located within a controlled access facility.	OE.PHYCAL Those responsible for the TOE must ensure that the TOE is protected from any physical attack.	OE.PHYCAL upholds this assumption by ensuring that the environment provides protection against physical attack.
A.MANAGE It is assumed that there are one or more competent individuals assigned to manage the TOE and the security of the information it contains.	OE.MANAGE Sites deploying the TOE will provide administrators for the TOE who are not careless, negligent, or willfully hostile, are appropriately trained and follow all administrator guidance. TOE administrators will ensure the system is used securely, including management of the audit trail.	OE.MANAGE upholds this assumption by ensuring that those responsible for the TOE will provide competent individuals to perform management of the security of the environment, and restrict these functions and facilities from unauthorized use.
	OE.REVIEW The configuration of the TOE will be inspected on a regular basis to ensure that the configuration continues to meet the organization's security policies in the face of: • Changes to the TOE configuration • Changes in the security	OE.REVIEW upholds this assumption by ensuring that administrators assigned to manage the TOE will review the configuration on a regular basis to ensure that it accurately reflects the intended configuration.

	objectives • Changes in the threats presented by the hostile network • Changes (additions and deletions) in the services available between the hostile network and the corporate network	
A.NOEVIL It is assumed that the users who manage the TOE are not careless, negligent, or willfully hostile, and follow all guidance.	OE.MANAGE Sites deploying the TOE will provide administrators for the TOE who are not careless, negligent, or willfully hostile, are appropriately trained and follow all administrator guidance. TOE administrators will ensure the system is used securely, including management of the audit trail.	OE.MANAGE upholds this assumption by ensuring that all administrators assigned to manage the TOE are not careless, negligent, or willfully hostile, are appropriately trained, and follow all administrator guidance.
A.FIREWALL It is assumed that all ports needed for proper operation of the TOE will be opened at the firewall. Also, any firewall settings necessary for the TOE's operation will be configured to allow the TOE to operate.	OE.FIREWALL The firewall must have all ports needed for proper operations of the TOE opened.	OE.FIREWALL upholds this assumption by ensuring that all ports necessary for the operation of the TOE are opened.
A.OS_AUTH It is assumed that the TOE environment will provide identification and authentication functions for users attempting to manage and use the TOE.	OE.OS_AUTH The operating system where the TOE is installed must provide authentication and identification of individuals attempting to use the TOE.	OE.OS_AUTH upholds this assumption by ensuring that the operating system where the TOE is installed will provide authentication and identification of users attempting to use the TOE.
A.SECCOMM It is assumed that the environment provides a sufficient level of protection to secure communications between distribution servers (if deployed), agents (if deployed) and other TOE components.	OE.SECCOMM The TOE environment must provide mechanisms to secure communications between TOE agents, distribution servers, and other TOE components.	OE.SECCOMM upholds this assumption by ensuring that the TOE environment will provide adequate security to protect the TOE.
A.FIPS A FIPS 140-2 validated cryptographic module in the TOE Environment must provide all cryptographic functionality for the TOE.	OE.FIPS The operating system that the TOE is installed upon must provide a FIPS 140-2 validated cryptographic module for the TOE to use to perform cryptographic functions.	OE.FIPS upholds this assumption by ensuring that a FIPS 140-2 cryptographic module is available for the TOE to use within the operating system the TOE is installed upon.

Every assumption is mapped to one or more Objectives in the table above. This complete mapping demonstrates that the defined security objectives uphold all defined assumptions.

8.3 Rationale for Extended Security Functional Requirements

A family of FDC requirements was created to specifically address the data collected and analyzed by patch and configuration management devices. The audit family of the CC (FAU) was used as a model for creating these requirements. The purpose of this family of requirements is to address the unique nature of patch deployments and configuration profiles and provide requirements about collecting, analyzing, storing, and reviewing the data. FDC_SCN.1 has no dependencies since the stated requirements embody all the necessary security functions. FDC_ANA.1 and FDC_STG.1 are dependent on FDC_SCN.1 since they apply to scan data that must first be collected by the TOE. These requirements exhibit functionality that can be easily documented in the ADV assurance evidence and thus do not require any additional Assurance Documentation.

FAU_GEN.1 (EXP) was created to address the audit data generation functionality of the TOE. FAU_GEN.1 was not chosen because the TOE does not explicitly log startup and shutdown of the audit function. By defining an explicit requirement FAU_GEN.1 (EXP) the Security Target can claim the audit functionality that the TOE supports.

8.4 Rationale for Extended TOE Security Assurance Requirements

There are no extended Security Assurance Requirements defined in this Security Target.

8.5 Security Requirements Rationale

The following discussion provides detailed evidence of coverage for each security objective.

8.5.1 Rationale for Security Functional Requirements of the TOE Objectives

Table 15 – Objectives:SFRs Mapping

Objective	Requirements Addressing the Objective	Rationale
O.LOG The TOE must record events of security relevance and provide authorized administrators with the ability to review the recorded events.	FAU_GEN.1 (EXP) Audit data generation	This requirement supports O.LOG by requiring the TOE to produce audit records for the system security events and for actions caused by enforcement of the Access Control, Protect, and Configure SFPs.
	FAU_SAR.1 Audit review	This requirement supports O.LOG by requiring the TOE to make the recorded audit records available for review.
O.MANAGE The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	FDP_ACC.1 Subset access control	This requirement supports O.MANAGE by requiring the TOE to enforce an access control policy on users connecting to the TOE.
	FDP_ACF.1 Security attribute based access control	This requirement supports O.MANAGE by defining the access control policy that controls interactions

		between users and the TOE.
	FMT_MOF.1 Management of security functions behaviour	This requirement supports O.MANAGE by defining the management functions available to each type of user.
	FMT_MSA.1a Management of security attributes (user roles)	This requirement supports O.MANAGE by restricting the users who can manage user roles.
	FMT_MSA.1b Management of security attributes (machine properties)	This requirement supports O.MANAGE by restricting the users who can manage machine groups.
	FMT_MSA.3a Static attribute initialisation (Access Control SFP)	This requirement supports O.MANAGE by defining restrictive default values for the Access Control policy.
	FMT_MSA.3b Static attribute initialisation (Protect SFP)	This requirement supports O.MANAGE by defining restrictive default values for the Protect policy.
	FMT_MSA.3c Static attribute initialization (Configure SFP)	This requirement supports O.MANAGE by defining restrictive default values for the Configure policy.
	FMT_MTD.1 Management of TSF data	This requirement supports O.MANAGE by restricting the users who can manage scanned data used for making security decisions.
O.MANAGE The TOE will provide all the functions and facilities necessary to support the administrators in their management of the security of the TOE, and restrict these functions and facilities from unauthorized use.	FMT_SMF.1 Specification of management functions	This requirement supports O.MANAGE by specifying the types of management functions available to users of the TOE.
	FMT_SMR.1 Security roles	This requirement supports O.MANAGE by specifying user roles and allowing the TOE to associate users with roles.
O.ROLE The TOE must be able to associate users and administrators with the appropriate role after the user or administrator authenticates.	FIA_ATD.1 User attribute definition	This requirement supports O.ROLE by requiring the TOE to maintain a list of user identifiers and their associated roles.
	FMT_SMR.1 Security roles	This requirement supports O.ROLE by requiring the TOE to be able to associate user roles with their respective users.
O.INTEGRITY The TOE must protect data being transmitted to physically separate	FPT_ITT.1 Basic internal TSF data transfer protection	This requirement supports O.INTEGRITY by requiring the TOE to protect TSF data from unauthorized modification while it is being

parts of the TOE from unauthorized modification.		transmitted between separate parts of the TOE.
	FPT_ITT.3 TSF data integrity monitoring	This requirement supports O.INTEGRITY by requiring the TOE to drop TSF data that has been modified or replaced by an unauthorized entity.
O.INT_ATK The TOE implementation must be able to mitigate attacks to stored executable code and thread overuse.	FPT_TST.1 TSF testing	This requirement supports O.INT_ATK by requiring the TOE to be able to perform a self test verifying the integrity of stored TOE executable code.
	FRU_RSA.1 Maximum quotas	This requirement supports O.INT_ATK by requiring the TOE to set a limit on the number of threads available for scanning machines simultaneously.
O.MONITOR The TOE must be able to monitor machines on the network to ensure that they exist in a secure state and alert TOE users if a system enters an insecure state.	FDP_IFC.1a Subset information flow control (Protect)	This requirement supports O.MONITOR by requiring the TOE to enforce the Protect SFP.
	FDP_IFF.1a Simple security attributes (Protect)	This requirement supports O.MONITOR by defining the attributes and information flow control rules for the Protect SFP.
	FDP_IFC.1b Subset information flow control (Configure)	This requirement supports O.MONITOR by requiring the TOE to enforce the Configure SFP.
	FDP_IFF.1b Simple security attributes (Configure)	This requirement supports O.MONITOR by defining the attributes and information flow control rules for the Configure SFP.
	FDC_ANA.1 (EXP) System Analysis	This requirement supports O.MONITOR by requiring the TOE to be able to analyze scanned data according to the Protect and Configure SFPs and alert administrators when security violations are discovered.
	FDC_SCN.1 (EXP) System Scan	This requirement supports O.MONITOR by requiring the TOE to be able to obtain system data from monitored machines.
	FDC_STG.1 (EXP) Scanned Data Storage	This requirement supports O.MONITOR by requiring the TOE to prevent unauthorized modification and deletion of scanned data.

8.5.2 Security Requirements Rationale for Refinement

This Security Target defines refinements to FTP_TST.1: TSF testing. These refinements were made because the TOE does not provide the ability for administrators to run self tests on the TOE executable code. Instead, the TOE automatically performs these integrity checks whenever a piece of TOE executable code is invoked.

8.5.3 Security Assurance Requirements Rationale

EAL3+ was chosen to provide a moderate level of assurance that is consistent with good commercial practices. As such, minimal additional tasks are placed upon the vendor assuming the vendor follows reasonable software engineering practices and can provide support to the evaluation for design and testing efforts. The chosen assurance level is appropriate with the threats defined for the environment. While the TOE may monitor a hostile environment, it is expected to be in a non-hostile position and embedded in or protected by other products designed to address threats that correspond with the intended environment. At EAL3+ the TOE will have incurred a search for obvious flaws to support its introduction into the non-hostile environment.

The augmentation of ALC_FLR.2 was chosen to give greater assurance of the developer's on-going flaw remediation process.

8.5.4 Dependency Rationale

This ST does satisfy all the requirement dependencies of the Common Criteria. Table 16 lists each requirement to which the TOE claims conformance with a dependency and indicates whether the dependent requirement was included. As the table indicates, all dependencies have been met.

Table 16 – Functional Requirements Dependencies

SFR ID	Dependencies	Dependency Met	Rationale
FAU_GEN.1 (EXP)	FPT_STM.1	No	Timestamps for the TOE are provided by the environment.
FAU_SAR.1	FAU_GEN.1 (EXP)	✓	
FDP_ACC.1	FDP_ACF.1	✓	
FDP_ACF.1	FDP_ACC.1	✓	
	FMT_MSA.3a	✓	
FDP_IFC.1a	FDP_IFF.1a	✓	
FDP_IFF.1a	FDP_IFC.1a	✓	
	FMT_MSA.3b	✓	
FDP_IFC.1b	FDP_IFF.1b	✓	
FDP_IFF.1b	FMT_MSA.3c	✓	
	FDP_IFC.1b	✓	

FIA_ATD.1	None	N/A	
FMT_MOF.1	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_MSA.1a	FDP_ACC.1	✓	
	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_MSA.1b	FMT_SMR.1	✓	
	FDP_IFC.1b	✓	
	FDP_IFC.1a	✓	
	FMT_SMF.1	✓	
FMT_MSA.3a	FMT_MSA.1a	✓	
	FMT_SMR.1	✓	
FMT_MSA.3b	FMT_MSA.1b	✓	
	FMT_SMR.1	✓	
FMT_MSA.3c	FMT_MSA.1b	✓	
	FMT_SMR.1	✓	
FMT_MTD.1	FMT_SMF.1	✓	
	FMT_SMR.1	✓	
FMT_SMF.1	None	N/A	
FMT_SMR.1	FIA_UID.1	No	Identification and authentication is provided by the operating system in the environment.
FPT_ITT.1	None	N/A	
FPT_ITT.3	FPT_ITT.1	✓	
FPT_TST.1	None	N/A	
FRU_RSA.1	None	N/A	
FDC_ANA.1 (EXP)	FDC_SCN.1 (EXP)	✓	
FDC_SCN.1 (EXP)	None	N/A	
FDC_STG.1 (EXP)	FDC_SCN.1 (EXP)	✓	

9 Acronyms

Table 17 – Acronyms

Acronym	Definition
CC	Common Criteria
CM	Configuration Management
EAL	Evaluation Assurance Level
ID	Identifier
IT	Information Technology
OS	Operating System
PP	Protection Profile
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SFP	Security Functional Policy
SSI	Shavlik Security Intelligence
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Function

Appendix A

This section lists the FIPS Certificate numbers for all versions of Windows used by the TOE.

FIPS Certificate #	Title	Software Version	File
869	Windows Server 2003 Kernel Cryptographic Module	5.2.3790.3959	fips.sys
1012	Windows Server 2003 Enhanced Cryptographic Provider	5.2.3790.4313	rsaenh.dll
989	Windows XP Enhanced Cryptographic Provider	5.1.2600.5507	rsaenh.dll
997	Microsoft Windows XP Kernel Mode Cryptographic Module	5.1.2600.5512	fips.sys
893	Windows Vista Enhanced Cryptographic Provider	6.0.6000.16386	rsaenh.dll
1010	Windows Server 2008 Enhanced Cryptographic Provider	6.0.6001.22202 and 6.0.6002.18005	rsaenh.dll