SonicWall SonicOS Enhanced V6.2 with IPS on NSA, SM, and TZ Appliances

Security Target

Doc No: 1962-000-D102
Version: 1.21P
12 January 2018

SonicWall, Inc.
5455 Great America Parkway,
Santa Clara, California, U.S.A.
95054

Prepared by:
EWA-Canada
1223 Michael Street North, Suite 200
Ottawa, Ontario, Canada
K1J7T2

CONTENTS

1 SECURITY TARGET INTRODUCTION 1
1.1 DOCUMENT ORGANIZATION 1
1.2 SECURITY TARGET REFERENCE 1
1.3 TOE REFERENCE 2
1.4 TOE OVERVIEW 2
1.5 TOE DESCRIPTION 2
1.5.1 Physical Scope 2
1.5.2 TOE Environment 3
1.5.3 TOE Guidance 4
1.5.4 Logical Scope 5
1.5.5 Functionality Excluded from the Evaluated Configuration 6
2 CONFORMANCE CLAIMS 7
2.1 COMMON CRITERIA CONFORMANCE CLAIM 7
2.2 ASSURANCE PACKAGE CLAIM 7
2.3 PROTECTION PROFILE CONFORMANCE CLAIM 7
3 SECURITY PROBLEM DEFINITION 9
3.1 THREATS 9
3.2 ORGANIZATIONAL SECURITY POLICIES 10
3.3 ASSUMPTIONS 11
4 SECURITY OBJECTIVES 13
4.1 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT 13
5 EXTENDED COMPONENTS DEFINITION 14
5.1 SECURITY FUNCTIONAL REQUIREMENTS 15
5.1.1 Class FAU: Security Audit 15
5.1.2 Class FCS: Cryptographic Support 17
5.1.3 Class FIA: Identification and Authentication 23
5.1.4 Class FPT: Protection of the TSF 27
5.1.5 Class FTA: TOE Access 31
5.2 SECURITY ASSURANCE REQUIREMENTS 32
6 SECURITY REQUIREMENTS 33

6.1 CONVENTIONS 33
6.2 TOE SECURITY FUNCTIONAL REQUIREMENTS 33
6.2.1 Security Audit (FAU) 35
6.2.2 Cryptographic Support (FCS) 38
6.2.3 Identification and Authentication (FIA) 44
6.2.4 Security Management (FMT) 46
6.2.5 Protection of the TSF (FPT) 47
6.2.6 TOE Access (FTA) 48
6.2.7 Trusted Path/Channels (FTP) 48
6.3 DEPENDENCY RATIONALE 49
6.4 TOE SECURITY ASSURANCE REQUIREMENTS 52
7 TOE SUMMARY SPECIFICATION 54
7.1 TOE SECURITY FUNCTIONS 54
7.1.1 Security Audit 54
7.1.2 Cryptographic Support 54
7.1.3 Identification and Authentication 65
7.1.4 Security Management 66
7.1.5 Protection of the TSF 66
7.1.6 TOE Access 68
7.1.7 Trusted Path / Channels 69
8 TERMINOLOGY AND ACRONYMS 70
8.1 TERMINOLOGY 70
8.2 ACRONYMS 70

LIST OF TABLES

Table 1 - TOE Appliances and Models 3
Table 2 – TOE Operational Environment Requirements 4
Table 3 - TOE Guidance Documentation 5
Table 4 – Logical Scope of the TOE 6
Table 5 – Threats 10
Table 6 – Organizational Security Policy 11
Table 7 – Assumptions 12
Table 8 – Security Objectives for the Operational Environment 13

Table 9 - Extended Security Functional Requirements 15
Table 10 – Summary of Security Functional Requirements 35
Table 11 – Security Functional Requirements and Auditable Events 38
Table 12 – Functional Requirement Dependencies 52
Table 13 – Security Assurance Requirements 53
Table 14 - Key Material 56
Table 15 - Cryptographic Functions 57
Table 16 - VPN Policies 62
Table 17 – Terminology 70
Table 18 – Acronyms 73

LIST OF FIGURES

Figure 1 – TOE Diagram 3
Figure 2 - Protected Audit Event Storage Component Leveling 16
Figure 3 - HTTPS Protocol Component Leveling 17
Figure 4 - IPsec Protocol Component Leveling 18
Figure 5 - Random Bit Generation Component Leveling 21
Figure 6 - TLS Server Protocol Component Leveling 22
Figure 7 - Password Management Component Leveling 23
Figure 8 - User Identification and Authentication Component Leveling 24
Figure 9 - Password-Based Authentication Mechanism Component Leveling 25
Figure 10 - Authentication Using X.509 Certificates Component Leveling 26
Figure 11 - Protection of Administrator Passwords Component Leveling 28
Figure 12 - Protection of TSF Data Component Leveling 28
Figure 13 - TSF Self Test Component Leveling 29
Figure 14 - Trusted Update Component Leveling 30
Figure 15 - TSF-Initiated Session Locking Component Leveling 32

1 SECURITY TARGET INTRODUCTION

This Security Target (ST) defines the scope of the evaluation in terms of the assumptions made, the intended environment for the TOE, the Information Technology (IT) security functional and assurance requirements to be met, and the level of confidence (evaluation assurance level) to which it is asserted that the TOE satisfies its IT security requirements. This document forms the baseline for the Common Criteria (CC) evaluation.

1.1 DOCUMENT ORGANIZATION

Section 1, ST Introduction, provides the Security Target (ST) reference, the Target of Evaluation (TOE) reference, the TOE overview and the TOE description.

Section 2, Conformance Claims, describes how the ST conforms to the Common Criteria, and Protection Profile.

Section 3, Security Problem Definition, describes the expected environment in which the TOE is to be used. This section defines the set of threats that are relevant to the secure operation of the TOE, organizational security policies with which the TOE must comply, and secure usage assumptions applicable to this analysis.

Section 4, Security Objectives, defines the set of security objectives to be satisfied by the TOE operating environment in response to the problem defined by the security problem definition.

Section 5, Extended Components Definition, defines the extended components which are then detailed in Section 6.

Section 6, Security Requirements, specifies the security functional and assurance requirements that must be satisfied by the TOE and the Information Technology (IT) environment.

Section 7, TOE Summary Specification, describes the security functions that are included in the TOE to enable it to meet the IT security functional requirements.

Section 8, Acronyms, defines the acronyms and terminology used in this ST.

1.2 SECURITY TARGET REFERENCE

ST Title:	SonicWall SonicOS Enhanced V6.2 with IPS on NSA, SM, and TZ Appliances Security Target
ST Version:	1.21P
ST Date:	12 January 2018

1.3 TOE REFERENCE

TOE Identification: SonicWall SonicOS Enhanced V6.2.5.0-50n on NSA, SM, and TZ Appliances

TOE Developer: SonicWall

TOE Type: Network Device

1.4 TOE OVERVIEW

The TOE is comprised of the SonicWall SonicOS Enhanced v6.2 software running on purpose built NSA, SM and TZ model hardware platforms.

The NSA, SM and TZ appliances support Virtual Private Network (VPN) functionality, which provides a secure connection between the device and the audit server. The appliances support authentication, and protect data from disclosure or modification during transfer.

The NSA, SM and TZ appliances are managed through a web based Graphical User Interface (GUI). All management activities may be performed through the web management GUI via a hierarchy of menu options. Administrators may configure policies and manage network traffic, users, and system logs.

1.5 TOE DESCRIPTION

1.5.1 Physical Scope

1.5.1.1 Physical Configuration

The TOE is a software and hardware TOE. It is a combination of a particular NSA, SM, or TZ hardware appliance and the SonicOS v6.2 software. Table 1 lists all the instances of the TOE that operate in the evaluated configuration. All listed TOE instances offer the same core functionality but vary in number of processors, physical size, and supported connections.

Appliance Series	Appliance Model
SonicWall NSA Series	NSA 2600
	NSA 3600
	NSA 4600
	NSA 5600
	NSA 6600
SonicWall SM Series	SM 9200
	SM 9400

Appliance Series	Appliance Model
	SM 9600
SonicWall TZ Series	TZ 300/W
	TZ 400/W
	TZ 500/W
	TZ 600

Table 1 - TOE Appliances and Models

In the evaluated configuration, the devices are placed in Network Device Protection Profile (NDPP) mode. NDPP mode is a configuration setting.

1.5.1.2 TOE Boundary

The SonicWall appliances are designed to filter traffic based on a set of rules created by a system administrator. The audit server provides a platform for sorting and viewing the log files that are produced by the appliance. Figure 1 below illustrates the physical boundary of the overall solution and ties together all of the administrative components of the TOE and the constituents of the operational environment.

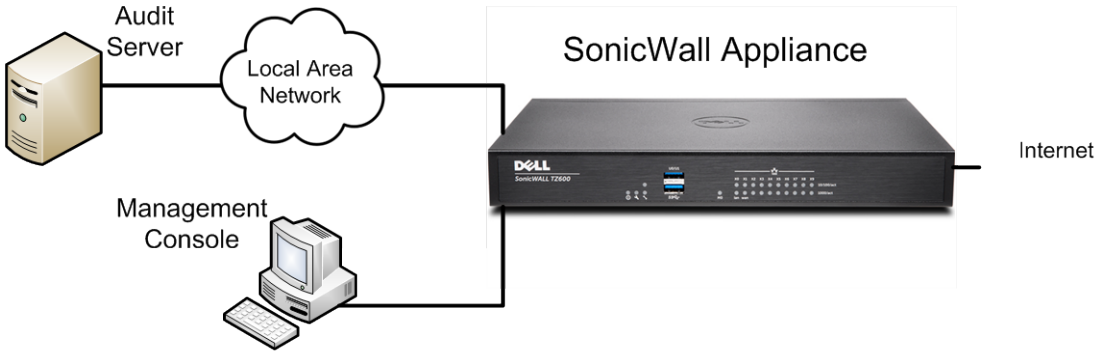

Figure 1 – TOE Diagram

1.5.2 TOE Environment

The following components are required for operation of the TOE in the evaluated configuration.

Component	Description / Requirements
Management Console	Any computer that provides a supported browser may be used to access the GUI. Firefox 47 was used in the

	evaluated configuration.
Audit Server	An event log server running on a general purpose computing platform that supports syslog (native to Ubuntu 10.04.3 LTS) with strongSwan (version 4.3.2-1.1ubuntu1) acting as the Ipsec VPN Client.

Table 2 – TOE Operational Environment Requirements

1.5.3 TOE Guidance

The TOE includes the following guidance documentation:

Document Type	Document Title
Quick Start Guides	SonicWall™ TZ300 / TZ300 Wireless Quick Start Guide 232-003785-50 Rev A Updated - January 2017
	SonicWall™ TZ400 / TZ400 Wireless Quick Start Guide 232-003783-50 Rev B Updated - February 2017
	SonicWall™ TZ500 / TZ500 Wireless Quick Start Guide 232-003781-50 Rev A Updated - January 2017
	SonicWall™ TZ600 Quick Start Guide 232-003779-50 Rev A Updated - January 2017
Getting Started Guides	SonicWall™ NSA 2600/3600/4600/5600/6600 Getting Started Guide 232-003419-51 Rev A Updated - March 2017
	SonicWall™ SuperMassive™ 9200/9400/9600 Getting Started Guide 232-000344-50 Rev A Updated - February 2017
Administration and Configuration Guides	SonicWall™ SonicOS 6.2 Administration Guide 232-002365-02 Rev B Updated - April 2017
	SonicWall™ SonicOS 6.2.5/6.2.7/6.2.9 Log Events Reference Guide 232-004020-00 Rev A Updated - July 2017

Document Type	Document Title
Common Criteria Guidance Supplement	SonicWALL SonicOS Enhanced V6.2 with IPS on NSA, SM, and TZ Appliances Common Criteria Guidance Supplement Version: 1.2, 10 May 2017

Table 3 - TOE Guidance Documentation

1.5.4 Logical Scope

The logical boundary of the TOE includes all interfaces and functions within the physical boundary. The logical boundary of the TOE may be broken down by the security function classes described in Section 6. Table 4 summarizes the logical scope of the TOE.

Functional Classes	Description
Security Audit	The TOE generates audit records for administrative activity, security related configuration changes, cryptographic key changes and startup and shutdown of the audit functions. The audit events are associated with the administrator who performs them, if applicable. The audit records are transmitted over an IPsec VPN tunnel to an external audit server in the IT environment for storage.
Cryptographic Support	The TOE provides cryptographic functions (key generation, key establishment, key destruction, cryptographic operation) to secure remote administrative sessions over Hypertext Transfer Protocol Secure (HTTPS)/Transport Layer Security (TLS), and the connection to the audit server using Internet Protocol Security (IPsec).
Identification and Authentication	The TOE provides a password-based logon mechanism. This mechanism enforces minimum strength requirements, and ensures that passwords are obscured when entered. The TOE also validates and authenticates X.509 certificates for all certificate use.
Security Management	The TOE provides management capabilities via a Web-Based GUI, accessed over HTTPS. Management functions allow the administrators to configure and update the system and manage users.
Protection of the TSF	The TOE prevents the reading of plaintext passwords and keys. The TOE provides a reliable timestamp for its own use. To protect the integrity of its security functions, the TOE implements a suite of self-tests at startup, and ensures that updates to the TOE software may be verified using a digital signature.

Functional Classes	Description
TOE Access	The TOE monitors local and remote sessions for inactivity and either locks or terminates the session when a threshold time period is reached. An advisory notice is displayed at the start of each session.
Trusted Path/Channels	The TSF provides IPsec VPN tunnels for trusted communication between itself and an audit server. The TOE implements HTTPS for protection of communications between itself and the Management Console.

Table 4 – Logical Scope of the TOE

1.5.5 Functionality Excluded from the Evaluated Configuration

The following features/functionality are excluded from this evaluation:

- Although SonicWall SonicOS Enhanced v6.2 supports several authentication mechanisms, the following mechanisms are excluded from the evaluated configuration:
  - Remote Authentication Dial-In User Service (RADIUS)
  - Lightweight Directory Access Protocol (LDAP)
  - Active Directory (AD)
  - eDirectory authentication
- Command Line Interface (CLI) (Secure Shell (SSH))
- Web Content Filtering
- Hardware Failover
- Real-time Blacklist (Simple Mail Transfer Protocol (SMTP))
- Global Security Client (including Group VPN)
- Global Management System
- SonicPoint
- Voice over IP (VoIP)
- Network Time Protocol (NTP)
- Antivirus
- Application Firewall
- IPS was not evaluated as part of the NDcPP evaluation

2 CONFORMANCE CLAIMS

2.1 COMMON CRITERIA CONFORMANCE CLAIM

This Security Target claims to be conformant to Version 3.1 of Common Criteria for Information Technology Security Evaluation according to:

- Common Criteria for Information Technology Security Evaluation, Part 1: Introduction and General Model; CCMB-2012-09-001, Version 3.1, Revision 4, September 2012
- Common Criteria for Information Technology Security Evaluation, Part 2: Security Functional Components; CCMB-2012-09-002, Version 3.1, Revision 4, September 2012
- Common Criteria for Information Technology Security Evaluation, Part 3: Security Assurance Components CCMB-2012-09-003, Version 3.1, Revision 4, September 2012

As follows:

- CC Part 2 extended
- CC Part 3 conformant

The Common Methodology for Information Technology Security Evaluation, Version 3.1, Revision 4, September 2012 has to be taken into account.

2.2 ASSURANCE PACKAGE CLAIM

This Security Target claims conformance to the Security Assurance Requirements (SARs) claimed in the collaborative Protection Profile for Network Devices (v 1.0, 27-Feb-2015).

2.3 PROTECTION PROFILE CONFORMANCE CLAIM

The TOE for this ST claims exact conformance with the collaborative Protection Profile for Network Devices (v1.0, 27-Feb-2015). The following Technical Decisions (TDs) also apply to this Security Target (as of 20 January 2017):

- TD0090: NIT Technical Decision for FMT_SMF.1.1 Requirement in NDcPP
- TD0093: NIT Technical Decision for FIA_X509_EXT.1.1 Requirement in NDcPP (revocation checking for TOE's own certificates during protocol negotiation requirement decision superseded by decision in TD0117)
- TD0094: NIT Technical Decision for validating a published hash in NDcPP
- TD0095: NIT Technical Interpretations regarding audit, random bit generation, and entropy in NDcPP
- TD0096: NIT Technical Interpretation regarding Virtualization
- TD0111: NIT Technical Decision for third party libraries and FCS_CKM.1 in NDcPP and FWcPP

- TD0112: NIT Technical Decision for TLS testing in the NDcPP v1.0 and FW cPP v1.0
- TD0113: NIT Technical Decision for testing and trusted updates in the NDcPP v1.0 and FW cPP v1.0
- TD0114: NIT Technical Decision for Re-Use of FIPS test results in NDcPP and FWcPP
- TD0115: NIT Technical Decision for Transport mode and tunnel mode in IPsec communication in NDcPP and FWcPP
- TD0116: NIT Technical Decision for a Typo in reference to RSASSA-PKCS1v1_5 in NDcPP and FWcPP
- TD0117: NIT Technical Decision for FIA_X509_EXT.1.1 Requirement in NDcPP
- TD0125: NIT Technical Decision for Checking validity of peer certificates for HTTPS servers
- TD0126: NIT Technical Decision for TLS Mutual Authentication
- TD0130: NIT Technical Decision for Requirements for Destruction of Cryptographic Keys
- TD0143: Failure testing for TLS session establishment in NDcPP and FWcPP

3 SECURITY PROBLEM DEFINITION

3.1 THREATS

Table 5 lists the threats addressed by the TOE.

Threat	Description
T.UNAUTHORIZED_ ADMINISTRATOR_ACCESS	Threat agents may attempt to gain administrator access to the network device by nefarious means such as masquerading as an administrator to the device, masquerading as the device to an administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between network devices. Successfully gaining administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_COMMUNICATION_ CHANNELS	Threat agents may attempt to target network devices that do not use standardized secure tunneling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the network device itself.
T.WEAK_AUTHENTICATION_ ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g., shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality

Threat	Description
	and integrity, and potentially the network device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the network device without administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_ COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials include replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or device credentials for use by the attacker.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.
T.SECURITY_FUNCTIONALITY_ FAILURE	A component of the network device may fail during start-up or during operations causing a compromise or failure in the security functionality of the network device, leaving the device susceptible to attackers.

Table 5 – Threats

3.2 ORGANIZATIONAL SECURITY POLICIES

Organizational Security Policies (OSPs) are security rules, procedures, or guidelines imposed on the operational environment. Table 6 lists the OSP that is presumed to be imposed upon the TOE or its operational environment by an organization that implements the TOE in the Common Criteria evaluated configuration.

OSP	Description
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

Table 6 – Organizational Security Policy

3.3 ASSUMPTIONS

The assumptions required to ensure the security of the TOE are listed in the following table:

Assumptions	Description
A.PHYSICAL_PROTECTION	The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.
A.LIMITED_FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example the device should not provide computing platform for general purpose applications (unrelated to networking functionality).
A.NO_THRU_TRAFFIC_ PROTECTION	A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs for particular types of network devices (e.g, firewall)
A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the network

Assumptions	Description
	device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious administrator that actively works to bypass or compromise the security of the device.
A.REGULAR_UPDATES	The network device firmware and software is assumed to be updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The administrator's credentials (private key) used to access the network device are protected by the platform on which they reside.

Table 7 – Assumptions

4 SECURITY OBJECTIVES

4.1 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT

The purpose of this section is to identify and describe the security objectives that are addressed by the operational environment. Table 8 identifies and describes these objectives.

Security Objective	Description
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_ PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.NO_THRU_TRAFFIC_ PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ ADMIN	TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner.
OE.UPDATES	The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_ SECURE	The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

Table 8 – Security Objectives for the Operational Environment

5 EXTENDED COMPONENTS DEFINITION

This section specifies the extended Security Functional Requirements (SFRs) used in this ST and defined in the PP. The definitions are taken directly from the collaborative Protection Profile for Network Devices, and are reproduced here without correction.

The following table identifies the extended SFRs that have been created to address additional security features of the TOE:

Class	Family	Component
FAU: Security Audit	FAU_STG_EXT: Security Audit Event Storage	FAU_STG_EXT.1: Protected Audit Event Storage
FCS: Cryptographic Support	FCS_HTTPS_EXT: HTTPS Protocol	FCS_HTTPS_EXT.1: HTTPS Protocol
	FCS_IPSEC_EXT: IPsec Protocol	FCS_IPSEC_EXT.1: Internet Protocol Security (IPsec) communications
	FCS_RBG_EXT: Random Bit Generation	FCS_RBG_EXT.1: Random Bit Generation
	FCS_TLSS_EXT: TLS Server Protocol	FCS_TLSS_EXT.1: TLS Server Protocol
FIA: Identification and Authentication	FIA_PMG_EXT: Password Management	FIA_PMG_EXT.1: Password Management
	FIA_UIA_EXT: User Identification and Authentication	FIA_UIA_EXT.1: User Identification and Authentication
	FIA_UAU_EXT: User Authentication	FIA_UAU_EXT.2: Password-Based Authentication Mechanism
	FIA_X509_EXT: Authentication Using X.509 Certificates	FIA_X509_EXT.1: Certificate Validation
		FIA_X509_EXT.2: Certification Authentication
		FIA_X509_EXT.3: Certificate Requests

Class	Family	Component
FPT: Protection of the TSF	FPT_APW_EXT: Protection of Administrator Passwords	FPT_APW_EXT.1: Protection of Administrator Passwords
	FPT_SKP_EXT: Protection of TSF Data	FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)
	FPT_TST_EXT: TSF Self Test	FPT_TST_EXT.1: TSF Testing
	FPT_TUD_EXT: Trusted Update	FPT_TUD_EXT.1: Trusted Update
FTA: TOE Access	FTA_SSL_EXT: TSF-Initiated Session Locking	FTA_SSL_EXT.1: TSF-Initiated Session Locking

Table 9 - Extended Security Functional Requirements

5.1 SECURITY FUNCTIONAL REQUIREMENTS

5.1.1 Class FAU: Security Audit

5.1.1.1 FAU_STG_EXT Protected Audit Event Storage

Family Behaviour

This component defines the requirements for the TSF to be able to securely transmit audit data between the TOE and an external IT entity.

Component Leveling

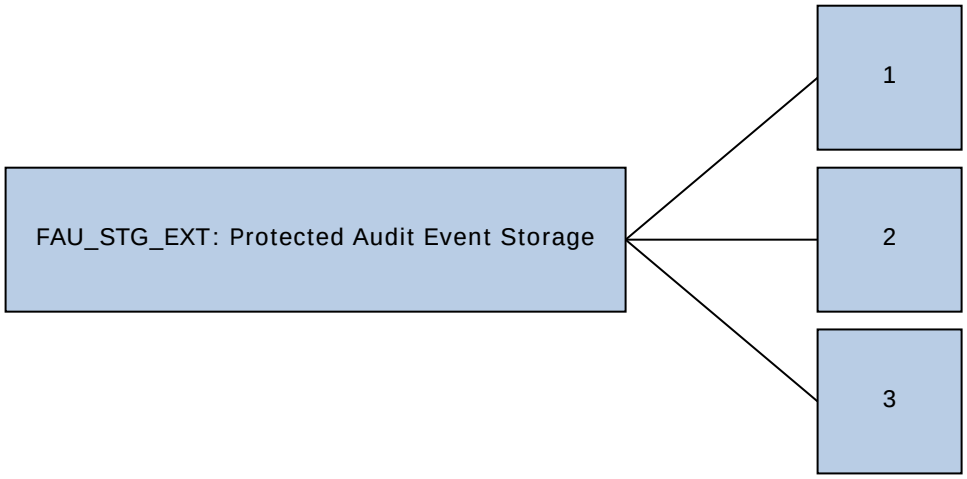

Figure 2 - Protected Audit Event Storage Component Leveling

FAU_STG_EXT.1 Protected audit event storage requires the TSF to use a trusted channel implementing a secure protocol.

FAU_STG_EXT.2 Counting lost audit data requires the TSF to provide information about audit records affected when the audit log becomes full.

FAU_STG_EXT.3 Display warning for local storage space requires the TSF to generate a warning before the audit log becomes full.

Note: FAU_STG_EXT.2 and FAU_STG_EXT.3 are not being claimed in this ST, therefore the extended component definitions have not been provided.

Management: FAU_STG_EXT.1

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

a) The TSF shall have the ability to configure the cryptographic functionality.

Audit: FAU_STG_EXT.1

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

a) No audit necessary.

FAU_STG_EXT.1 Protected Audit Event Storage

Hierarchical to:	No other components
Dependencies:	FAU_GEN.1 Audit data generation
	FTP_ITC.1 Inter-TSF Trusted Channel

FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself.

FAU_STG_EXT.1.3 The TSF shall [selection: drop new audit data, overwrite previous audit records according to the following rule: [assignment: *rule for overwriting previous audit records*], [*assignment: other action*]] when the local storage space for audit data is full.

5.1.2 Class FCS: Cryptographic Support

5.1.2.1 FCS_HTTPS_EXT HTTPS Protocol

Family Behaviour

Components in this family define the requirements for protecting remote management sessions between the TOE and a Security Administrator. This family describes how HTTPS will be implemented. This is a new family defined for the FCS Class.

Component Leveling

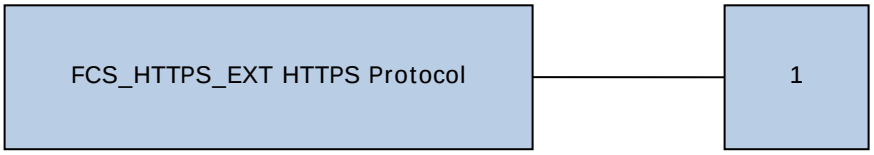

Figure 3 - HTTPS Protocol Component Leveling

FCS_HTTPS_EXT.1 HTTPS requires that HTTPS be implemented according to RFC 2818 and supports TLS.

Management: FCS_HTTPS_EXT.1

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

a) There are no management activities foreseen.

Audit: FCS_HTTPS_EXT.1

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

a) There are no auditable events foreseen.

FCS_HTTPS_EXT.1	**HTTPS Protocol**
Hierarchical to:	No other components
Dependencies:	FCS_TLS_EXT.1 TLS Protocol
FCS_HTTPS_EXT.1.1	The TSF shall implement the HTTPS protocol that complies with RFC 2818.
FCS_HTTPS_EXT.1.2	The TSF shall implement the HTTPS protocol using TLS.
FCS_HTTPS_EXT.1.3	The TSF shall establish the connection only if [selection: the peer presents a valid certificate during handshake, the peer initiates handshake].

5.1.2.2 FCS_IPSEC_EXT IPsec Protocol

Family Behaviour

Components in this family address the requirements for protecting communications using IPsec.

Component Leveling

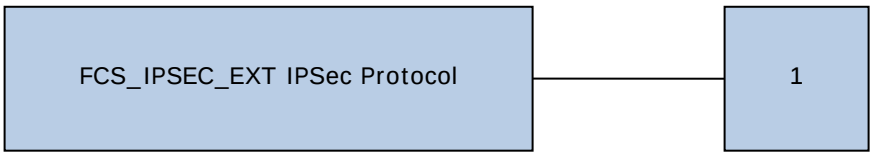

Figure 4 - IPsec Protocol Component Leveling

FCS_IPSEC_EXT.1 IPsec requires that IPsec be implemented as specified.

Management: FCS_IPSEC_EXT.1

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

a) Maintenance of SA lifetime configuration.

Audit: FCS_IPSEC_EXT.1

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

a) Decisions to DISCARD, BYPASS, PROTECT network packets processed by the TOE.
b) Failure to establish an IPsec SA
c) IPsec SA establishment
d) IPsec SA termination
e) Negotiation “down” from an IKEv2 to IKEv1 exchange.

FCS_IPSEC_EXT.1	**Internet Protocol Security (IPsec) Communications**	
	Hierarchical to:	No other components
	Dependencies:	FCS_CKM.1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption) FCS_COP.1(2) Cryptographic operation (Signature Verification) FCS_COP.1(3) Cryptographic operation (Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation
FCS_IPSEC_EXT.1.1	The TSF shall implement the IPsec architecture as specified in RFC 4301.	
FCS_IPSEC_EXT.1.2	The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.	

FCS_IPSEC_EXT.1.3 The TSF shall implement transport mode and [selection: tunnel mode, no other mode].

FCS_IPSEC_EXT.1.4 The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-CBC-128, AES-CBC-256 (both specified by RFC 3602) and [selection: AES-GCM-128 (specified in RFC 4106), AES-GCM-256 (specified in RFC 4106), no other algorithms] together with a Secure Hash Algorithm (SHA)-based HMAC.

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [selection:

- IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [selection: no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions];
- IKEv2 as defined in RFCs 5996 [selection: with no support for NAT traversal, with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)], and [selection: no other RFCs for hash functions, RFC 4868 for hash functions]].

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [selection: IKEv1, IKEv2] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 3602 and [selection: AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm].

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [selection:

- IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on [selection:

  o number of bytes;

  o length of time, where the time values can configured within [assignment: *integer range including 24*] hours;];

- IKEv2 SA lifetimes can be configured by a Security Administrator based on [selection:

  o number of bytes;

  o length of time, where the time values can configured within [assignment: *integer range including 24*] hours]

].

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [selection:

- IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on [selection:

  o number of bytes;

o length of time, where the time values can be configured within [assignment: *integer range including 8*] hours;];

- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [selection:

  o number of bytes;

  o length of time, where the time values can be configured within [assignment: *integer range including 8*] hours;]

].

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [assignment: *(one or more) number(s) of bits that is at least twice the security strength of the negotiated Diffie-Hellman group*] bits.

FCS_IPSEC_EXT.1.10 The TSF shall generate nonces used in [selection: IKEv1, IKEv2] exchanges of length [selection:

- [assignment: *security strength associated with the negotiated Diffie-Hellman group*];
- at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash].

FCS_IPSEC_EXT.1.11 The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and [selection: 19 (256-bit Random ECP), 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), [assignment: *other DH groups that are implemented by the TOE*], no other DH groups].

FCS_IPSEC_EXT.1.12 The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: IKEv1 Phase 1, IKEv2 IKE_SA] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: IKEv1 Phase 2, IKEv2 CHILD_SA] connection.

FCS_IPSEC_EXT.1.13 The TSF shall ensure that all IKE protocols perform peer authentication using [selection: RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [selection: Pre-shared Keys, no other method].

FCS_IPSEC_EXT.1.14 The TSF shall only establish a trusted channel to peers with valid certificates.

5.1.2.3 FCS_RBG_EXT Random Bit Generation

Family Behaviour

Components in this family address the requirements for random bit/number generation. This is a new family define do for the FCS class.

Component Leveling

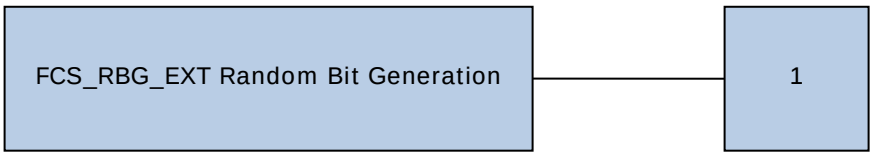

Figure 5 - Random Bit Generation Component Leveling

FCS_RBG_EXT.1 Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source.

Management: FCS_RBG_EXT.1

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

a) There are no management activities foreseen.

Audit: FCS_RBG_EXT.1

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

a) Minimal: failure of the randomization process

FCS_RBG_EXT.1 Random Bit Generation

Hierarchical to: No other components

Dependencies: No other components

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [selection: [assignment: *number of software-based sources*] software-based noise source, [assignment: *number of hardware-based sources*] hardware-based noise source] with minimum of [selection; 128 bits, 192 bits, 256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

5.1.2.4 FCS_TLSS_EXT TLS Server Protocol

Family Behaviour

The component in this family addresses the ability for a server to use TLS to protect data between a client and the server using the TLS protocol.

Component Leveling

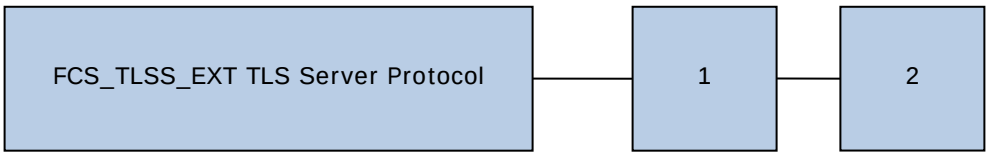

Figure 6 - TLS Server Protocol Component Leveling

FCS_TLSS_EXT.1 TLS Server requires that the server side of TLS be implemented as specified.

FCS_TLSS_EXT.2 TLS Server requires the mutual authentication be included in the TLS implementation.

Note: FCS_TLSS_EXT.2 is not being claimed in this ST, therefore the extended component definition has not been provided.

Management: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

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

a) There are no management activities foreseen.

Audit: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

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

a) Failure of TLS session establishment.

b) TLS session establishment

c) TLS session termination

FCS_TLSS_EXT.1 TLS Server Protocol

Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FCS_COP.1(1) Cryptographic Operation (AES Data encryption/decryption FCS_COP.1(2) Cryptographic Operation (Signature Verification) FCS_COP.1(3) Cryptographic Operation (Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSS_EXT.1.1 The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

- Mandatory Ciphersuites:

  - o [assignment: *List of mandatory ciphersuites and reference to RFC in which each is defined*]
- [selection: Optional Ciphersuites:
  - o [assignment: *List of optional ciphersuites and reference to RFC in which each is defined*]
  - o no other ciphersuite]].

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, and [selection: TLS 1.1, TLS 1.2, none].

FCS_TLSS_EXT.1.3 The TSF shall generate key establishment parameters using RSA with key size 2048 bits and [selection: 3072 bits, 4096 bits, no other size] and [selection: [assignment: *List of elliptic curves*]; [assignment: *List of diffie-hellman parameter sizes*]].

5.1.3 Class FIA: Identification and Authentication

5.1.3.1 FIA_PMG_EXT Password Management

Family Behaviour

The TOE defines the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained.

Component Leveling

Figure 7 - Password Management Component Leveling

FIA_PMG_EXT.1 Password management requires the TSF to support passwords with varying composition requirements, minimum lengths, maximum lifetime, and similarity constraints.

Management: FIA_PMG_EXT.1

No management functions.

Audit: FIA_PMG_EXT.1

No specific audit requirements.

FIA_PMG_EXT.1 Password Management

Hierarchical to: No other components

Dependencies: No other components

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following

special characters: [selection: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [assignment: *other characters*]];

b) Minimum password length shall be settable by the Security Administrator, and support passwords of 15 characters or greater.

5.1.3.2 FIA_UIA_EXT User Identification and Authentication

Family Behaviour

The TSF allows certain specified actions before the non-TOE entity goes through the identification and authentication process.

Component Leveling

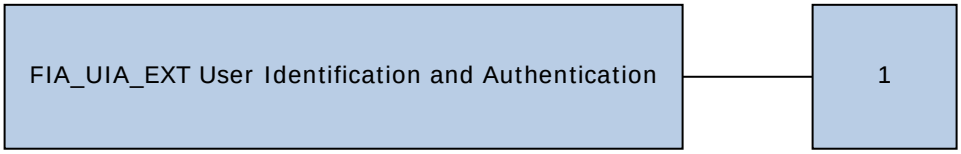

Figure 8 - User Identification and Authentication Component Leveling

FIA_UIA_EXT.1 User Identification and Authentication requires administrators (including remote administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path. It also ensures that every user is identified and authenticated before the TOE performs any mediated functions

Management: FIA_UIA_EXT.1

The following actions could be considered for the management functions in FMT:
a) Ability to configure the list of TOE services available before an entity is identified and authenticated

Audit: FIA_UIA_EXT.1

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

a) All use of the identification and authentication mechanism

b) Provided user identity, origin of the attempt (e.g. IP address)

FIA_UIA_EXT.1 User Identification and Authentication

Hierarchical to: No other components

Dependencies: FTA_TAB.1 Default TOE Access Banners

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non- TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;

- [selection: no other actions, [assignment: *list of services, actions performed by the TSF in response to non-TOE requests.*]]

FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

5.1.3.3 FIA_UAU_EXT User Authentication

Family Behaviour

Provides for a locally based administrative user authentication mechanism.

Component Leveling

Figure 9 - Password-Based Authentication Mechanism Component Leveling

FIA_UAU_EXT.2 The password-based authentication mechanism provides administrative users a locally based authentication mechanism.

Management: FIA_UAU_EXT.2

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

a) None

Audit: FIA_UAU_EXT.2

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

a) Minimal: All use of the authentication mechanism.

FIA_UAU_EXT.2 Password-based Authentication Mechanism

Hierarchical to: No other components

Dependencies: None

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [selection: [assignment: *other authentication mechanism(s)*], none] to perform administrative user authentication.

5.1.3.4 FIA_X509_EXT Authentication Using X.509 Certificates

Family Behaviour

This family defines the behaviour, management, and use of X.509 certificates for functions to be performed by the TSF. Components in this family require validation of certificates according to a specified set of rules, use of certificates

for authentication for protocols and integrity verification, and the generation of certificate requests.

Component Leveling

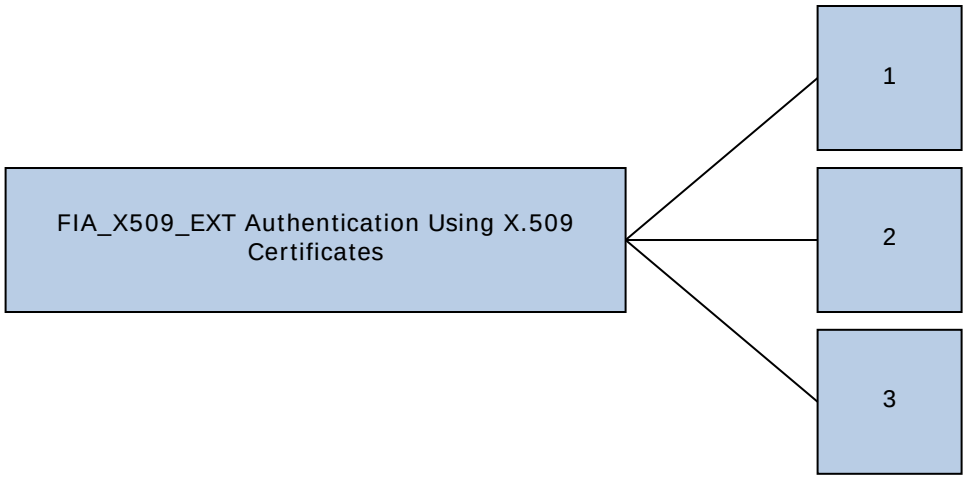

Figure 10 - Authentication Using X.509 Certificates Component Leveling

FIA_X509_EXT.1 X509 Certificate Validation, requires the TSF to check and validate certificates in accordance with the RFCs and rules specified in the component.

FIA_X509_EXT.2 X509 Certificate Authentication, requires the TSF to use certificates to authenticate peers in protocols that support certificates, as well as for integrity verification and potentially other functions that require certificates.

FIA_X509_EXT.3 X509 Certificate Requests, requires the TSF to be able to generate Certificate Request Messages and validate responses.

Management: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

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

a) Remove imported X.509v3 certificates
b) Approve import and removal of X.509v3 certificates
c) Initiate certificate requests

Audit: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

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

a) Minimal: No specific audit requirements are specified.

FIA_X509_EXT.1 X.509 Certificate Validation

Hierarchical to: No other components

Dependencies: No other components

FIA_X509_EXT.1.1 The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.

- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [selection: the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759].
- The TSF shall validate the extendedKeyUsage field according to the following rules: [assignment: *rules that govern contents of the extendedKeyUsage field that need to be verified*].

FIA_X509_EXT.1.2 The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

FIA_X509_EXT.2 X.509 Certificate Authentication

Hierarchical to: No other components

Dependencies: No other components

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection: IPsec, TLS, HTTPS, SSH, [assignment: other protocols], no protocols], and [selection: code signing for system software updates, code signing for integrity verification, [assignment: *other uses*], no additional uses].

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate].

FIA_X509_EXT.3 X.509 Certificate Requests

Hierarchical to: No other components

Dependencies: No other components

FIA_X509_EXT.3.1 The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [selection: device-specific information, Common Name, Organization, Organizational Unit, Country, [assignment: *other information*]].

FIA_X509_EXT.3.2 The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.1.4 Class FPT: Protection of the TSF

5.1.4.1 FPT_APW_EXT Protection of Administrator Passwords

Family Behaviour

Components in this family ensure that the TSF will protect plaintext credential data such as passwords from unauthorized disclosure.

Component Leveling

Figure 11 - Protection of Administrator Passwords Component Leveling

FPT_APW_EXT.1 Protection of administrator passwords requires that the TSF prevent plaintext credential data from being read by any user or subject.

Management: FPT_APW_EXT.1

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

a) No management functions.

Audit: FPT_APW_EXT.1

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

a) No audit necessary.

FPT_APW_EXT.1	**Protection of Administrator Passwords**
Hierarchical to:	No other components
Dependencies:	No other components

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

5.1.4.2 FPT_SKP_EXT Protection of TSF Data

Family Behaviour

Components in this family address the requirements for managing and protecting TSF data, such as cryptographic keys. This is a new family modelled after the FPT_PTD[1] Class.

Component Leveling

Figure 12 - Protection of TSF Data Component Leveling

[1] This class does not appear in CC Part 2.

FPT_SKP_EXT.1 Protection of TSF Data (for reading all symmetric keys), requires preventing symmetric keys from being read by any user or subject. It is the only component of this family.

Management: FPT_SKP_EXT.1

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

a) There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

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

a) There are no auditable events foreseen.

FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

Hierarchical to: No other components

Dependencies: No other components

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.1.4.3 FPT_TST_EXT TSF Self Test

Family Behaviour

Components in this family address the requirements for self-testing the TSF for selected correct operation.

Component Leveling

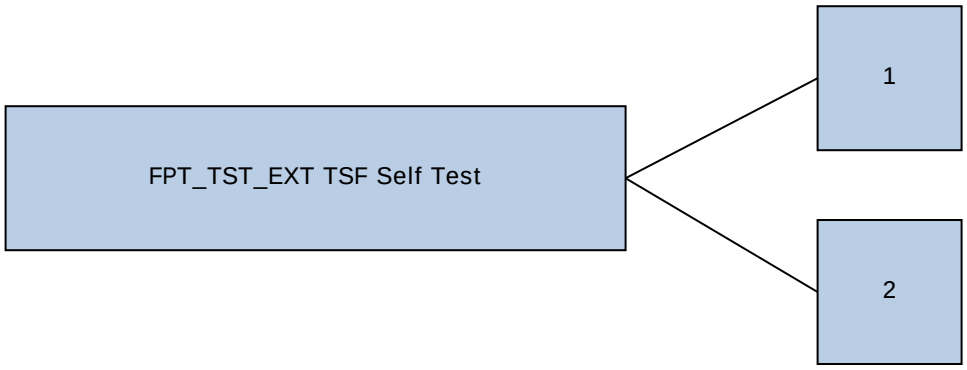

Figure 13 - TSF Self Test Component Leveling

FPT_TST_EXT.1 TSF Self Test requires a suite of self tests to be run during initial start-up in order to demonstrate correct operation of the TSF.

FPT_TST_EXT.2 Self tests based on certificates applies when using certificates as part of self test, and requires that the self test fails if a certificate is invalid

Note: FPT_TST_EXT.2 is not being claimed in this ST, therefore the extended component definition has not been provided.

Management: FPT_TST_EXT.1

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

a) No management functions

Audit: FPT_TST_EXT.1

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

a) Indication that TSF self test was completed

FPT_TST_EXT.1 TSF Testing

Hierarchical to: No other components

Dependencies: None

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [selection: during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: *conditions under which self-tests should occur*]] to demonstrate the correct operation of the TSF: [assignment: *list of self-tests run by the TSF*].

5.1.4.4 FPT_TUD_EXT Trusted Update

Family Behaviour

Components in this family address the requirements for updating the TOE firmware and/or software.

Component Leveling

Figure 14 - Trusted Update Component Leveling

FPT_TUD_EXT.1 Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.

Management: FPT_TUD_EXT.1

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

a) Ability to update the TOE and to verify the updates

b) Ability to update the TOE and to verify the updates using the digital signature capability (FCS_COP.1(2)) and [selection: no other functions, [assignment: other cryptographic functions (or other functions) used to support the update capability]]
c) Ability to update the TOE, and to verify the updates using [selection: digital signature, published hash, no other mechanism] capability prior to installing those updates

Audit: FPT_TUD_EXT.1

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

a) Initiation of the update process.
b) Any failure to verify the integrity of the update

FPT_TUD_EXT.1	**Trusted Update**
Hierarchical to:	No other components
Dependencies:	FCS_COP.1(1) Cryptographic Operation (for cryptographic signature), or FCS_COP.1(3) Cryptographic Operation (for cryptographic hashing)

FPT_TUD_EXT.1.1 The TSF shall provide [assignment: *authorised users*] the ability to query the currently executing version of the TOE firmware/software as well as the most recently installed version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide [assignment: *authorised users*] the ability to manually initiate updates to TOE firmware/software and [selection: support automatic checking for updates, support automatic updates, no other update mechanism].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [selection: digital signature mechanism, published hash] prior to installing those updates.

5.1.5 Class FTA: TOE Access

5.1.5.1 FTA_SSL_EXT TSF-Initiated Session Locking

Family Behaviour

Components in this family address the requirements for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

Component Leveling

Figure 15 - TSF-Initiated Session Locking Component Leveling

FTA_SSL_EXT.1 TSF-initiated session locking, requires system initiated locking of an interactive session after a specified period of inactivity. It is the only component of this family.

Management: FTA_SSL_EXT.1

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

a) Specification of the time of user inactivity after which lock-out occurs for an individual user.

Audit: FTA_SSL_EXT.1

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

a) Any attempts at unlocking an interactive session.

FTA_SSL_EXT.1 TSF-Initiated Session Locking

Hierarchical to: No other components

Dependencies: FIA_UAU.1 Timing of Authentication

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [selection:

- lock the session – disable any activity of the user's data access/display devices other than unlocking the session, and requiring that the administrator re-authenticate to the TSF prior to unlocking the session;
- terminate the session]

after a Security Administrator-specified time period of inactivity.

5.2 SECURITY ASSURANCE REQUIREMENTS

This ST does not include extended Security Assurance Requirements.

6 SECURITY REQUIREMENTS

This section provides security functional and assurance requirements that must be satisfied by a compliant TOE. These requirements consist of functional components from Part 2 of the CC, extended requirements, and an Evaluation Assurance Level (EAL) that contains assurance components from Part 3 of the CC.

6.1 CONVENTIONS

The CC permits four types of operations to be performed on functional requirements: selection, assignment, refinement, and iteration. These operations, when performed on requirements that derive from CC Part 2, are identified in this ST in the following manner:

- Selection: Indicated by surrounding brackets, e.g., [selected item].
- Assignment: Indicated by surrounding brackets and italics, e.g., [*assigned item*]. Selections within assignments are also in italics.
- Refinement: Refined components are identified by using **bold** for additional information, or ~~strikeout~~ for deleted text.
- Iteration: Indicated by assigning a number in parenthesis to the end of the functional component identifier as well as by modifying the functional component title to distinguish between iterations, e.g., 'FDP_ACC.1(1), Subset access control (administrators)' and 'FDP_ACC.1(2) Subset access control (devices)'.

6.2 TOE SECURITY FUNCTIONAL REQUIREMENTS

The security functional requirements for this ST consist of the following components from Part 2 of the CC and extended components defined in Section 5, summarized in Table 10 - Summary of Security Functional Requirements.

Class	Identifier	Name
Security Audit (FAU)	FAU_GEN.1	Audit data generation
	FAU_GEN.2	User identity association
	FAU_STG_EXT.1	Protected audit event storage
Cryptographic Support (FCS)	FCS_CKM.1	Cryptographic key generation
	FCS_CKM.2	Cryptographic key establishment
	FCS_CKM.4	Cryptographic key Destruction
	FCS_COP.1(1)	Cryptographic operation (AES Data Encryption/Decryption)

Class	Identifier	Name
	FCS_COP.1(2)	Cryptographic operation (Signature Generation and Verification)
	FCS_COP.1(3)	Cryptographic operation (Hash algorithm)
	FCS_COP.1(4)	Cryptographic operation (Keyed Hash Algorithm)
	FCS_HTTPS_EXT.1	HTTPS protocol
	FCS_IPSEC_EXT.1	IPSec protocol
	FCS_RBG_EXT.1	Random bit generation
	FCS_TLSS_EXT.1	TLS server protocol
Identification and Authentication (FIA)	FIA_PMG_EXT.1	Password management
	FIA_UIA_EXT.1	User identification and authentication
	FIA_UAU_EXT.2	Password-based authentication mechanism
	FIA_UAU.7	Protected authentication feedback
	FIA_X509_EXT.1	X.509 certificate validation
	FIA_X509_EXT.2	X.509 certificate authentication
	FIA_X509_EXT.3	X.509 certificate requests
Security Management (FMT)	FMT_MOF.1(1)/ TrustedUpdate	Management of security functions behaviour
	FMT_MTD.1	Management of TSF data
	FMT_SMF.1	Specification of management functions
	FMT_SMR.2	Restrictions on security roles

Class	Identifier	Name
Protection of the TSF (FPT)	FPT_APW_EXT.1	Protection of administrator passwords
	FPT_SKP_EXT.1	Protection of TSF data (for reading of all symmetric keys)
	FPT_STM.1	Reliable time stamps
	FPT_TST_EXT.1	TSF testing
	FPT_TUD_EXT.1	Trusted update
TOE Access (FTA)	FTA_SSL_EXT.1	TSF-initiated session locking
	FTA_SSL.3	TSF-initiated termination
	FTA_SSL.4	User-initiated termination
	FTA_TAB.1	Default TOE access banners
Trusted path/channels (FTP)	FTP_ITC.1	Inter-TSF trusted channel
	FTP_TRP.1	Trusted path

Table 10 – Summary of Security Functional Requirements

6.2.1 Security Audit (FAU)

6.2.1.1 FAU_GEN.1 Audit data generation

Hierarchical to: No other components.

Dependencies: FPT_STM.1 Reliable time stamps

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

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

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

c) [*All administrative actions comprising:*

- *Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).*
- *Security related configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).*
- *Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).*
- *Resetting passwords (name of related user account shall be logged).*

- *Starting and stopping services (if applicable)*
- *[*no other actions*]*;

d) *Specifically defined auditable events listed in Table 11.*

].

FAU_GEN.1.2 The TSF shall record within each audit record at least the following information:

a) Date and time of the event, type of event, subject identity (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, [*information specified in column three of Table 11*].

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1(1)	None.	None.
FCS_COP.1(2)	None.	None.
FCS_COP.1(3)	None.	None.
FCS_COP.1(4)	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session	Reason for failure
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA	Reason for failure
FCS_RBG_EXT.1	None.	None.
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and	Provided user identity, origin of

Requirement	Auditable Events	Additional Audit Record Contents
	authentication mechanism.	the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1(1)/ TrustedUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1	All management activities of TSF data.	None.
FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	No additional information.
FPT_STM.1	Changes to time.	The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	Identification of the claimed user identity.

Table 11 – Security Functional Requirements and Auditable Events

6.2.1.2 FAU_GEN.2 User identity association

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FIA_UID.1 Timing of identification

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

6.2.1.3 FAU_STG_EXT.1 Protected Audit Event Storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FTP_ITC.1 Inter-TSF Trusted Channel

FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself.

FAU_STG_EXT.1.3 The TSF shall [overwrite previous audit records according to the following rule: [*new records overwrite the oldest records*]] when the local storage space for audit data is full.

6.2.2 Cryptographic Support (FCS)

6.2.2.1 FCS_CKM.1 Cryptographic key generation

Hierarchical to: No other components.

Dependencies: [FCS_CKM.2 Cryptographic key establishment, or
FCS_COP.1 Cryptographic operation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.1.1 The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm [

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;*
- *FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1*

] ~~and specified cryptographic key sizes [assignment: *cryptographic key sizes*] that meet the following: [assignment: *list of standards*]~~.

6.2.2.2 FCS_CKM.2 Cryptographic key establishment

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_CKM.2.1 The TSF shall ~~distribute~~ **perform** cryptographic ~~keys~~ **key establishment** in accordance with a specified cryptographic key ~~distribution~~ **establishment** method [

- *RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography";*
- *Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"*

] ~~that meets the following: [assignment: *list of standards*]~~.

6.2.2.3 FCS_CKM.4 Cryptographic key destruction

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method [

- *For plaintext keys in volatile storage, the destruction shall be executed by a [single overwrite consisting of* [*a pseudo-random pattern using the TSF's RBG*];
- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that* [

- o *Logically addresses the storage location of the key and performs a* [*single-pass*] *overwrite consisting of* [*a pseudo-random pattern using the TSF's RBG*]]

] that meets the following: [*No Standard*].

6.2.2.4 FCS_COP.1(1) Cryptographic operation (AES Data Encryption / Decryption)

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(1) The TSF shall perform [*encryption/decryption*] in accordance with a specified cryptographic algorithm [*AES used in* [*CBC*] *mode*] and cryptographic key sizes [*128, 256 bits*] that meet the following: [*AES as specified in ISO 18033-3,* [*CBC as specified in ISO 10116*]].

6.2.2.5 FCS_COP.1(2) Cryptographic operation (Signature Generation and Verification)

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(2) The TSF shall perform [*cryptographic signature services (generation and verification)*] in accordance with a specified cryptographic algorithm [

- *RSA Digital Signature Algorithm*] and cryptographic key sizes **(modulus)** [*2048 bits*]

that meets the following: [

For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA- PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3].

6.2.2.6 FCS_COP.1(3) Cryptographic operation (Hash Algorithm)

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security

attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(3) The TSF shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm [*SHA-1, SHA-256, SHA-384, SHA-512*] ~~and cryptographic key sizes [assignment: *cryptographic key sizes*]~~ that meet the following: [*ISO/IEC 10118-3:2004*].

6.2.2.7 FCS_COP.1(4) Cryptographic operation (Keyed Hash Algorithm)

Hierarchical to: No other components.

Dependencies: [FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation]
FCS_CKM.4 Cryptographic key destruction

FCS_COP.1.1(4) The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm [*HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512*] and cryptographic key sizes [*512, 1024*] **and message digest sizes [160, 256, 384, 512] bits** that meet the following: [*ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*].

6.2.2.8 FCS_HTTPS_EXT.1 HTTPS protocol

Hierarchical to: No other components

Dependencies: FCS_TLS_EXT.1 TLS Protocol

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3 The TSF shall establish the connection only if [the peer initiates the handshake].

6.2.2.9 FCS_IPSEC_EXT.1 IPSec protocol

Hierarchical to: No other components

Dependencies: FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (Signature Verification)
FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_IPSEC_EXT.1.1 The TSF shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2 The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

FCS_IPSEC_EXT.1.3 The TSF shall implement transport mode and [no other mode].

FCS_IPSEC_EXT.1.4 The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-CBC-128, AES-CBC-256 (both specified by RFC 3602) and [no other algorithms] together with a Secure Hash Algorithm (SHA)-based HMAC.

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [

- IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [no other RFCs for extended sequence numbers], and [no other RFCs for hash functions]].

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [IKEv1] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 3602 and [no other algorithm].

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [

- IKEv1 Phase 1 SA lifetimes can be configured by an Security Administrator based on [
  - length of time, where the time values can configured within [*2 minutes to 24*] hours]].

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [

- IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on [
  - length of time, where the time values can be configured within [*2 minutes to 8*] hours]].

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [180, 224, 256, 384] bits.

FCS_IPSEC_EXT.1.10 The TSF shall generate nonces used in [IKEv1] exchanges of length [

- [*90, 112, 128, 192*]].

FCS_IPSEC_EXT.1.11 The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and [5 (1536-bit MODP), 19 (256-bit Random ECP), 20 (384-bit Random ECP)].

FCS_IPSEC_EXT.1.12 The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [IKEv1 Phase 1] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [IKEv1 Phase 2] connection.

FCS_IPSEC_EXT.1.13 The TSF shall ensure that all IKE protocols perform peer authentication using [RSA] that use X.509v3 certificates that conform to RFC 4945 and [no other method].

FCS_IPSEC_EXT.1.14 The TSF shall only establish a trusted channel to peers with valid certificates.

6.2.2.10 FCS_RBG_EXT.1 Random bit generation

Hierarchical to: No other components

Dependencies: No other components

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [Hash_DRBG (any)].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[*four*] software-based noise source, [*one*] hardware- based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

6.2.2.11 FCS_TLSS_EXT.1 TLS server protocol

Hierarchical to: No other components

Dependencies: FCS_CKM.1 Cryptographic Key Generation
FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (Signature Verification)
FCS_COP.1(3) Cryptographic operation (Hash Algorithm)
FCS_RBG_EXT.1 Random bit generation

FCS_TLSS_EXT.1.1 The TSF shall implement [TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

- Mandatory Ciphersuites:
  - [*TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*]
- [Optional Ciphersuites:
  - [*TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
  - *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
  - *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246*]

].

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, and [none].

FCS_TLSS_EXT.1.3 The TSF shall generate key establishment parameters using RSA with key size 2048 bits and [no other size] and [Diffie-Hellman parameters of size 2048 bits and [no other size]].

6.2.3 Identification and Authentication (FIA)

6.2.3.1 FIA_PMG_EXT.1 Password management

Hierarchical to: No other components.

Dependencies: No other components.

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [*no other characters*]];

b) Minimum password length shall be settable by the Security Administrator, and shall support passwords of 15 characters or greater.

6.2.3.2 FIA_UIA_EXT.1 User identification and authentication

Hierarchical to: No other components.

Dependencies: FTA_TAB.1 Default TOE Access Banners

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [no other actions]

FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

6.2.3.3 FIA_UAU_EXT.2 Password-based authentication mechanism

Hierarchical to: No other components.

Dependencies: None

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [none] to perform administrative user authentication.

6.2.3.4 FIA_UAU.7 Protected authentication feedback

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FIA_UAU.7.1 The TSF shall provide only [*obscured feedback*] to the administrative user while the authentication is in progress.

6.2.3.5 FIA_X509_EXT.1 X.509 Certificate validation

Hierarchical to: No other components.

Dependencies: No other components

FIA_X509_EXT.1.1 The TSF shall validate certificates in accordance with the following rules:

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 2560].
- The TSF shall validate the extendedKeyUsage field according to the following rules: [
  - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
  - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field*.]

FIA_X509_EXT.1.2 The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

6.2.3.6 FIA_X509_EXT.2 X.509 Certificate authentication

Hierarchical to: No other components
Dependencies: No other components

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [IPsec], and [no additional uses].

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [not accept the certificate].

6.2.3.7 FIA_X509_EXT.3 X.509 Certificate requests

Hierarchical to: No other components
Dependencies: No other components

FIA_X509_EXT.3.1 The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [Common Name, Organization, Country].

FIA_X509_EXT.3.2 The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

6.2.4 Security Management (FMT)

6.2.4.1 FMT_MOF.1(1)/TrustedUpdate Management of security functions behaviour

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MOF.1.1(1) The TSF shall restrict the ability to [enable] the functions [*perform manual update*] to [*Security Administrators*].

6.2.4.2 FMT_MTD.1 Management of TSF data

Hierarchical to: No other components.

Dependencies: FMT_SMR.1 Security roles

FMT_SMF.1 Specification of Management Functions

FMT_MTD.1.1 The TSF shall restrict the ability to [[*manage*]] the [*TSF data*] to [*Security Administrators*].

6.2.4.3 FMT_SMF.1 Specification of management functions

Hierarchical to: No other components.

Dependencies: No dependencies.

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using* [*digital signature*] *capability prior to installing those updates;*]
- [
  - o *Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1;*
  - o *Ability to configure the cryptographic functionality*].

6.2.4.4 FMT_SMR.2 Restrictions on security roles

Hierarchical to: FMT_SMR.1 Security roles

Dependencies: FIA_UID.1 Timing of identification

FMT_SMR.2.1 The TSF shall maintain the roles: [*Security Administrator*].

FMT_SMR.2.2 The TSF shall be able to associate users with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions [

- *The Security Administrator role shall be able to administer the TOE locally;*
- *The Security Administrator role shall be able to administer the TOE remotely*]

are satisfied.

6.2.5 Protection of the TSF (FPT)

6.2.5.1 FPT_APW_EXT.1 Protection of administrator passwords

Hierarchical to: No other components

Dependencies: No other components.

FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

6.2.5.2 FPT_SKP_EXT.1 Protection of TSF data (for reading of all symmetric keys)

Hierarchical to: No other components.

Dependencies: No other components.

FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

6.2.5.3 FPT_STM.1 Reliable time stamps

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

6.2.5.4 FPT_TST_EXT.1 TSF testing

Hierarchical to: No other components.

Dependencies: No dependencies.

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [during initial start-up (on power on)] to demonstrate the correct operation of the TSF: [

- *Appliance Power on self-test consisting of a CPU and RAM test*
- *Firmware integrity test (using 16-bit CRC EDC)*
- *AES-CBC Encrypt and Decrypt Known Answer Tests*
- *SHA-1, -256, -384, -512 Known Answer Tests*
- *HMAC-SHA-1, -256, -512 Known Answer Tests*
- *DSA Signature Verification Pairwise Consistency Test*
- *RSA Sign and Verify Known Answer Tests*
- *DH Pairwise Consistency Test*
- *DRBG Known Answer Test*

].

6.2.5.5 FPT_TUD_EXT.1 Trusted update

Hierarchical to: No other components.

Dependencies: FCS_COP.1(1) Cryptographic operation (for cryptographic signature), or
FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)

FPT_TUD_EXT.1.1 The TSF shall provide [*Security Administrators*] the ability to query the currently executing version of the TOE firmware/software as well as the most recently installed version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide [*Security Administrators*] the ability to manually initiate updates to TOE firmware/software and [no other update mechanism].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [digital signature mechanism] prior to installing those updates.

6.2.6 TOE Access (FTA)

6.2.6.1 FTA_SSL_EXT.1 TSF-initiated session locking

Hierarchical to: No other components.

Dependencies: FIA_UAU.1 Timing of authentication

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [terminate the session] after a Security Administrator-specified time period of inactivity.

6.2.6.2 FTA_SSL.3 TSF-initiated termination

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_SSL.3.1 The TSF shall terminate ~~an~~ **a remote** interactive session after a [*Security Administrator-configurable time interval of session inactivity*].

6.2.6.3 FTA_SSL.4 User-initiated termination

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_SSL.4.1 The TSF shall allow ~~user~~ **Administrator**-initiated termination of the ~~user's~~ **Administrator's** own interactive session.

6.2.6.4 FTA_TAB.1 Default TOE access banners

Hierarchical to: No other components.

Dependencies: No dependencies.

FTA_TAB.1.1 Before establishing a**n administrative** user session, the TSF shall display ~~an~~ **a Security Administrator-specified** advisory **notice and consent** warning message regarding unauthorised use of the TOE.

6.2.7 Trusted Path/Channels (FTP)

6.2.7.1 FTP_ITC.1 Inter-TSF trusted channel

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_ITC.1.1 The TSF shall **be capable of using [IPsec] to** provide a communication channel between itself and **authorized** ~~another trusted~~ IT **entities supporting the following capabilities: audit server, [[*no other capabilities*]]** ~~product~~ that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from ~~modification or~~ disclosure **and detection of modification of the channel data**.

FTP_ITC.1.2 The TSF shall permit [the TSF, or the authorized IT entities] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [*transmission of audit data*].

6.2.7.2 FTP_TRP.1 Trusted path

Hierarchical to: No other components.

Dependencies: No dependencies.

FTP_TRP.1.1 The TSF shall **be capable of using [TLS/HTTPS] to** provide a communication path between itself and **authorized** [remote] ~~users~~ **administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [disclosure, [*and provides detection of modification of the channel data*]].

FTP_TRP.1.2 The TSF shall permit [remote ~~users~~ **administrators**] to initiate communication via the trusted path.

FTP_TRP.1.3 The TSF shall require the use of the trusted path for [initial ~~user~~ **administrator** authentication, [*and all remote administration actions*]].

6.3 DEPENDENCY RATIONALE

Table 12 identifies the SFRs from Part 2 of the CC, extended SFRs identified in Section 5, and their associated dependencies. It also indicates whether the ST explicitly addresses each dependency.

SFR	Dependency	Dependency Satisfied	Rationale
FAU_GEN.1	FPT_STM.1	✓	
FAU_GEN.2	FAU_GEN.1	✓	
	FIA_UID.1	✓	The dependency is satisfied by FIA_UID_EXT.1, which covers the identification requirement for the collaborative Protection Profile for Network Devices.
FAU_STG_EXT.1	FAU_GEN.1	✓	
	FTP_ITC.1	✓	

SFR	Dependency	Dependency Satisfied	Rationale
FCS_CKM.1	FCS_CKM.2 or FCS_COP.1	✓	Satisfied by FCS_COP.1
	FCS_CKM.4	✓	
FCS_CKM.2	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	✓	Satisfied by FCS_CKM.1
	FCS_CKM.4	✓	
FCS_CKM.4	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	✓	Satisfied by FCS_CKM.1
FCS_COP.1(1)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	No	This dependency is not met in the claimed Protection Profile. It may be considered to be resolved by the protocol utilizing the session algorithms.
	FCS_CKM.4	✓	
FCS_COP.1(2)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	✓	Satisfied by FCS_CKM.1
	FCS_CKM.4	✓	
FCS_COP.1(3)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	No	FCS_COP.1(3) has no key management dependencies in the claimed Protection Profile.
	FCS_CKM.4	✓	
FCS_COP.1(4)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	No	This dependency is not met in the claimed Protection Profile. It may be considered to be resolved by the protocol utilizing the session algorithms.
	FCS_CKM.4	✓	
FCS_HTTPS_ EXT.1	FCS_TLSS_ EXT.1	✓	

SFR	Dependency	Dependency Satisfied	Rationale
FCS_IPSEC_ EXT.1	FCS_CKM.1	✓	
	FCS_CKM.2	✓	
	FCS_COP.1(1)	✓	
	FCS_COP.1(2)	✓	
	FCS_COP.1(3)	✓	
	FCS_RBG_ EXT.1	✓	
FCS_RBG_EXT.1	None	N/A	
FCS_TLSS_EXT.1	FCS_CKM.1	✓	
	FCS_COP.1(1)	✓	
	FCS_COP.1(2)	✓	
	FCS_COP.1(3)	✓	
	FCS_RBG_ EXT.1	✓	
FIA_PMG_EXT.1	None	N/A	
FIA_UIA_EXT.1	FTA_TAB.1	✓	
FIA_UAU_EXT.2	None	N/A	
FIA_UAU.7	FIA_UAU.1	✓	The dependency is satisfied by FIA_UAU_EXT.2, which covers the authentication requirement for the collaborative Protection Profile for Network Devices.
FIA_X509_EXT.1	None	N/A	
FIA_X509.EXT.2	None	N/A	
FIA_X509.EXT.3	None	N/A	
FMT_MOF.1(1)/ TrustedUpdate	FMT_SMR.1	✓	The dependency is satisfied by FMT_SMR.2, which is hierarchical to FMT_SMR.1.
	FMT_SMF.1	✓	
FMT_MTD.1	FMT_SMR.1	✓	The dependency is satisfied by FMT_SMR.2, which is hierarchical to FMT_SMR.1.

SFR	Dependency	Dependency Satisfied	Rationale
	FMT_SMF.1	✓	
FMT_SMF.1	None	N/A	
FMT_SMR.2	FIA_UID.1	✓	The dependency is satisfied by FIA_UIA_EXT.1, which covers the identification requirement for the collaborative Protection Profile for Network Devices.
FPT_APW_EXT.1	None	N/A	
FPT_SKP_EXT.1	None	N/A	
FPT_STM.1	None	N/A	
FPT_TST_EXT.1	None	N/A	
FPT_TUD_EXT.1	FCS_COP.1(1) or FCS_COP.1(3)	✓	The dependency is satisfied by FCS_COP.1 which covers cryptographic signature.
FTA_SSL_EXT.1	FIA_UAU.1	✓	The dependency is satisfied by FIA_UAU_EXT.2, which covers the authentication requirement for the collaborative Protection Profile for Network Devices.
FTA_SSL.3	None	N/A	
FTA_SSL.4	None	N/A	
FTA_TAB.1	None	N/A	
FTP_ITC.1	None	N/A	
FTP_TRP.1	None	N/A	

Table 12 – Functional Requirement Dependencies

6.4 TOE SECURITY ASSURANCE REQUIREMENTS

The TOE security assurance requirements for this ST conform to those described in the claimed Protection Profile.

The security assurance requirements are summarized in the following table:

Assurance Class	Assurance Components	
	Identifier	**Name**
Development (ADV)	ADV_FSP.1	Basic functional specification
Guidance Documents (AGD)	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
Life-Cycle Support (ALC)	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Security Target Evaluation (ASE)	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.1	Security objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
Tests (ATE)	ATE_IND.1	Independent testing - conformance
Vulnerability Assessment (AVA)	AVA_VAN.1	Vulnerability survey

Table 13 – Security Assurance Requirements

Note that the following refinement has been made to ASE_TSS.1.1:

ASE_TSS.1.1C The TOE summary specification shall describe how the TOE meets each SFR. **In the case of entropy analysis the TSS is used in conjunction with, including required supplementary information on Entropy.**

7 TOE SUMMARY SPECIFICATION

This section provides a description of the security functions and assurance measures of the TOE that meet the TOE security requirements.

7.1 TOE SECURITY FUNCTIONS

A description of each of the TOE security functions follows.

7.1.1 Security Audit

The TOE generates audit records and stores them as management logs and user activity logs. The management logs record administrative logins and management activity, including changes to configuration and access control policies. User activity logs record blocked traffic, blocked websites, and VPN activity. Each record contains the date and time, event type, subject identity (when applicable) and outcome of the event. For events caused by a user, the identity of the user is included in the audit record.

Contents of all audit records are described in the SonicOS Log Events Reference Guide. This includes administrator login and management activities associated with cryptographic keys. The logs do not contain the cryptographic keys.

In the evaluated configuration, the TOE is configured to send audit records to an audit server over an IPsec protected link. The link is established between the TOE and the audit server, and the records are sent over this connection. The logs are sent continuously, and are removed from the buffer as they are sent. If the connection to the audit server is lost, the logs are stored in a 32 kilobyte rolling log buffer. When the buffer becomes full, the oldest logs are overwritten. When contained on the TOE, the logs are stored in a specifically reserved area of the System RAM. Access to these records is restricted to authorized administrators with the appropriate privilege and cannot be modified or deleted. Users who do not have the required privilege are not able to access the audit records.

TOE Security Functional Requirements addressed: FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1.

7.1.2 Cryptographic Support

7.1.2.1 Cryptographic Key Generation and Key Establishment

The TOE supports Rivest-Shamir-Adleman (RSA) using cryptographic 2048-bit key sizes and Finite Field Cryptography (FFC) schemes using cryptographic 2048-bit key sizes. RSA is used in support of TLS. FFC is used in support of IPsec.

ECDSA is used to verify the digital signature on firmware updates (CAVP # 1315).

TOE Security Functional Requirements addressed: FCS_CKM.1, FCS_CKM.2.

7.1.2.2 Cryptographic Key Destruction

The TOE does not support any plaintext key material. All keys, including public keys and shared secrets, are stored encrypted. Key materials held in volatile and non-volatile memory are zeroized after use by direct overwrite consisting of a pseudo-random pattern. The overwrites are read and verified.

The table below shows the origin, storage location and destruction details for all plaintext keys. Unless otherwise stated, the keys are generated by the TOE.

Type / Description	Generation / Algorithm	Storage	Destruction Method
RSA private key used for TLS	RSA (2048 bits)	Stored encrypted in flash memory Held in the RAM buffer in plaintext	The encrypted key is overwritten with a block erase when deleted The plaintext key is overwritten with a pseudo-random pattern upon termination of the session or reboot of the appliance
RSA public key used for TLS	RSA (2048 bits)	Stored encrypted in flash memory Held in the RAM buffer in plaintext	The encrypted key is overwritten with a block erase when deleted The plaintext key is overwritten with a pseudo-random pattern upon termination of the session or reboot of the appliance
AES key used for TLS	AES-128 AES-256	Keys are not stored Held in the RAM buffer in plaintext	The key is overwritten with a pseudo-random pattern upon termination of the session or reboot of the appliance
DH Keys used for TLS	DH (2048 bits)	Keys are not stored Held in the RAM buffer in plaintext	The keys are overwritten with a pseudo-random pattern upon termination of the session or reboot of the appliance
DH Keys used for IPsec	DH (2048 bits)	Stored encrypted in System RAM Held in the RAM buffer in plaintext	The encrypted key is overwritten with a pseudo-random pattern upon termination of the session or reboot of the appliance The plaintext key is

Type / Description	Generation / Algorithm	Storage	Destruction Method
			overwritten with a pseudo-random pattern upon termination of the session or reboot of the appliance
AES Keys used for IPsec	AES-128 AES-256	Keys are not stored Held in the RAM buffer in plaintext	The plaintext key is overwritten with a pseudo-random pattern upon termination of the session or reboot of the appliance
SonicWall Public Key used to verify firmware updates	ECDSA (p-256 NIST curve)	Stored encrypted in Flash Memory	The key may be overwritten by a software update

Table 14 - Key Material

The TOE includes two types of memory: RAM and flash. Ephemeral keys are only held in RAM, either in the System RAM or the RAM buffer. The RAM buffer is an area of the System RAM that is allocated for data storage for a period of time. Private keys are only held in plaintext in the RAM buffer. Private keys and public key certificates are stored encrypted in flash memory using OpenSSL 1.0.1. Private and public keys are overwritten in the RAM buffer after use.

Setting the TOE to factory default erases all keys, including those stored in the flash memory.

TOE Security Functional Requirements addressed: FCS_CKM.4.

7.1.2.3 Cryptographic Operation

Cryptographic support is provided by cryptographic algorithms within the TOE devices. The applicable Cryptographic Algorithm Validation Program (CAVP) certificate numbers associated with the claimed functions are shown in Table 15.

Function / Algorithm	Details	CAVP Certificate
Encryption/decryption using AES in CBC mode	128, 256 bit key sizes	5070
RSA digital signature algorithm	2048 bit key size	2750
SHA-1, SHA-256, SHA-384, and SHA-512 Cryptographic hashing	160 bit message digest for SHA-1 256 bit message digest for SHA-256 384 bit message digest for SHA-384	4130

Function / Algorithm	Details	CAVP Certificate
services	512 bit message digest for SHA-512	
HMAC-SHA-1 Keyed hash message authentication	Key length: 512 bit Hash function: SHA-1 Block size: 512 bit Output MAC length: 160 bit	3384
HMAC-SHA-256 Keyed hash message authentication	Key length: 512 bit Hash function: SHA-256 Block size: 512 bit Output MAC length: 256 bit	3384
HMAC-SHA-384 Keyed hash message authentication	Key length: 1024 bit Hash function: SHA-384 Block size: 1024 bit Output MAC length: 384 bit	3384
HMAC-SHA-512 Keyed hash message authentication	Key length: 1024 bit Hash function: SHA-512 Block size: 1024 bit Output MAC length: 512 bit	3384
Key establishment according to NIST SP 800-56A Rev 2	FCC Schemes with SHA-256	CVL 1631
Key derivation according to NIST SP 800-135 Rev 1	IKE v1 IKE v2 TLS 1.1	CVL 1632 (KDF135)
ECDSA Signature verification	P-256 curve	1315
DRBG	Mode: SHA-256	1887

Table 15 - Cryptographic Functions

The TOE supports signature verification for ECDSA, in accordance with FIPS PUB 186-4, implementing a P-256 NIST curve. ECDSA signature generation is not supported by the TOE.

The TOE provides cryptographic hashing services for key generation using SHA-256 as specified in NIST SP 800-90 DRBG. SHA-1 and SHA-256 are used for TLS. SHA-1, SHA-256, SHA-384, and SHA-512 are used for IPsec.

The TOE implements a NIST SP 800-56B section 8.2 conformant RSA-based key establishment scheme for asymmetric key establishment. SHA-1 and SHA-256 are used for secure hashing and RSA is used for digital signatures. SHA-256 is used with ECDSA for the verification of firmware.

Within the TLS implementation, the claimed cryptographic algorithms are used in support of all of the supported ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256

TOE Security Functional Requirements addressed: FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4).

7.1.2.4 IPsec

The TOE Administrator implements an IPsec policy to encrypt data between the TOE and the audit server.

In general, an IPsec policy may be established to encrypt data (PROTECT). If traffic not belonging to the protected interface or subnet is found on this interface, the traffic will bypass encryption and be routed to the destination in plaintext (BYPASS). If plaintext traffic is received on a protected interface or subnet, the traffic is discarded and deleted (DISCARD).

This section describes IPsec rule configuration and processing. Note that when the TOE device is placed in NDPP mode, only the collaborative Protection Profile for Network Devices (NDcPP) allowed algorithms are supported and visible to the administrator. NDPP mode is a configuration setting.

IPsec VPN traffic is secured in two stages:

- Authentication: The first phase establishes the authenticity of the sender and receiver of the traffic using an exchange of the public key portion of a public-private key pair. This phase must be successful before the VPN tunnel can be established.
- Encryption: The traffic in the VPN tunnel is encrypted using AES.

The exchange of information to authenticate the members of the VPN and encrypt/decrypt the data uses the Internet Key Exchange (IKE) protocol for exchanging authentication information (keys) and establishing the VPN tunnel.

7.1.2.4.1 IKE Version 1

IKE version 1 uses a two phase process to secure the VPN tunnel.

- IKE Phase 1 is the authentication phase. The nodes or gateways on either end of the tunnel authenticate with each other, exchange encryption/decryption keys, and establish the secure tunnel.
- IKE Phase 2 is the negotiation phase. Once authenticated, the two nodes or gateways negotiate the methods of encryption and data verification (using a hash function) to be used on the data passed through the VPN and negotiate the number of security associations (SAs) in the tunnel and their lifetime before requiring renegotiation of the encryption/decryption keys.

7.1.2.4.1.1 IKE Phase 1

In IKE v1, there are two modes of exchanging authentication information:

- Main Mode: The node or gateway initiating the VPN queries the node or gateway on the receiving end, and they exchange authentication methods, public keys, and identity information. This usually requires six messages back and forth. The order of authentication messages in Main Mode is:
  1. The initiator sends a list of cryptographic algorithms the initiator supports.
  2. The responder replies with a list of supported cryptographic algorithms.
  3. The initiator sends a public key (part of a Diffie-Hellman public/private key pair) for the first mutually supported cryptographic algorithm.
  4. The responder replies with the public key for the same cryptographic algorithm.
  5. The initiator sends identity information (a certificate).
  6. The responder replies with identity information.
- Aggressive mode is not supported in the evaluated configuration.

7.1.2.4.1.2 IKE Phase 2

In IKE phase 2, the two parties negotiate the type of security to use, which encryption methods to use for the traffic through the tunnel (if needed), and negotiate the lifetime of the tunnel before re-keying is needed.

The two types of security for individual packets are:

- Encapsulating Security Payload (ESP), in which the data portion of each packet is encrypted using a protocol negotiated between the parties.
- Authentication Header (AH), in which the header of each packet contains authentication information to ensure the information is authenticated and has not been tampered with. No encryption is used for the data with AH.

SonicOS supports the following encryption algorithms to protect traffic as it passes through the VPN tunnel: AES-128 and AES-256.

7.1.2.4.2 Security Policy Database

The TOE administrative interface provides a VPN Policies page on which the policies applicable to a particular VPN may be displayed. This page has four tabs (General, Proposals, Advanced, Client) to enter the appropriate rules. The rules for processing both inbound and outbound packets are determined by these policies.

7.1.2.4.2.1 VPN Policies

The following table shows the policy options that may be selected when configuring the VPN Policies.

Tab	Policy	Options
General (Security	Policy Type	Site to Site

Tab	Policy	Options
Policy)	Authentication Method	IKE using 3rd party certificates
	Name	Name given to the VPN
	IPsec Primary Gateway Name or Address	Host name or IP address of the remote connection
	IPsec Secondary Gateway Name or Address	Used if the remote device supports more than one endpoint
	Gateway Certificate	The certificate is selected from the list of installed certificates
General (IKE Authentication)	Local IKE ID (type and address)	Optional. IPv4 Address (default) Domain Name Email AddressDevice Identifier Key Identifier
	Peer IKE ID (type and address)	Optional. IPv4 Address (default) Domain Name Email AddressDevice Identifier Key Identifier
Network (Local Networks)	Select one of: Use this VPN Tunnel as default route for all Internet traffic or Choose destination network from list	If a specific local network can access the VPN tunnel, select a local network from the Choose local network from list drop-down menu. If traffic can originate from any local network, select Any Address. Use this option if a peer has 'Use this VPN tunnel as default route for all Internet traffic' selected.
Network (Remote Networks)	Select one of: Choose local network from list or Any address	If traffic from any local user cannot leave the appliance unless it is encrypted, select ‘Use this VPN Tunnel as default route for all Internet traffic’. Alternatively, select Choose Destination network from list, and select the address object or group.
Proposals (IKE Phase 1)	Exchange	Main Mode, IKEv2 Mode
	DH Group	Group 5, Group 14, 256-bit Random ECP Group or 384-bit Random ECP Group.
	Encryption	AES-128, AES-256
	Authentication	SHA-256

Tab	Policy	Options
	Life Time (seconds)	120 to 86400 seconds
Proposals (IPsec Phase 2)	Protocol	Encapsulating Security Protocol (ESP)
	Encryption	AES-128, AES-256
	Authentication	SHA-256
	Enable Perfect Forward Secrecy (checkbox)	When selected, an additional Diffie-Hellman key exchange is performed
	Life Time (seconds)	120 to 28800 seconds
Advanced (Advanced Settings – Main Mode options)	Enable Keep Alive	Select Enable Keep Alive to use heartbeat messages between peers on this VPN tunnel. If one end of the tunnel fails, this will initiate automatic renegotiation of the tunnel once both sides become available.
	Suppress automatic Access Rules creation for VPN Policy	Not enabled by default
	Disable IPsec Anti-Replay	Stops packets with duplicate sequence numbers from being dropped
	Require authentication of VPN Clients by XAUTH	This is used to require XAUTH authentication by users prior to allowing traffic to traverse this tunnel. If selected, the appropriate user group must be identified.
	Enable Windows Networking (NetBIOS) broadcast	Allows access to remote network resources to browse the Windows Network
	Enable Multicast	Enables IP multicasting traffic, such as streaming audio (including VoIP) and video applications, to pass through the VPN tunnel
	Permit Acceleration	Enables redirection of traffic matching this policy to a WAN Acceleration (WXA) appliance.
	Apply NAT policies	To perform Network Address Translation on the Local Network, select or create an Address Object in the Translated Local Network menu. To translate the Remote Network, select or create an Address Object in the Translated Remote Network

Tab	Policy	Options
		drop-down menu.
	Management via this SA	If using the VPN policy to manage the appliance, select HTTPS as the management method.
	User login via this SA	Select HTTP or HTTPS; however, HTTP is not allowed with remote authentication.
	Default LAN Gateway (optional)	This option may be used if a router is used on the LAN for traffic entering this tunnel destined for an unknown subnet. For example, if the remote connection is configured to Use this VPN Tunnel as default route for all Internet traffic, enter the IP address of the router into the Default LAN Gateway (optional) field.
	VPN Policy bound to	Select an interface or zone from the drop down menu.

Table 16 - VPN Policies

TOE Security Functional Requirements addressed: FCS_IPSEC_EXT.1.1, FCS_IPSEC_EXT.1.2.

7.1.2.4.3 Transport Mode

The TOE can be only operated in Transport mode in the evaluated configuration. This is set using the 'Advanced' tab of the VPN Settings menu.

TOE Security Functional Requirements addressed: FCS_IPSEC_EXT.1.3.

7.1.2.4.4 AES and HMAC Implementation

AES-CBC-128 and AES-CBC-256 are supported for IPsec. The HMAC implementation conforms to HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512.

TOE Security Functional Requirements addressed: FCS_IPSEC_EXT.1.4, FCS_IPSEC_EXT.1.6.

7.1.2.4.5 IKEv1

IKEv1 is supported. Main mode is used for IKEv1. Aggressive mode is not used for IKEv1.

TOE Security Functional Requirements addressed: FCS_IPSEC_EXT.1.5

7.1.2.4.6 Lifetime Configuration Method

The IKEv1 Phase 1 SA lifetime is selected in the SPD and may be set to be between 120 and 86400 seconds (24 hours). The IKEv1 Phase 2 SA lifetime is

selected in the SPD and may also be set to be between 120 and 28800 seconds (8 hours).

TOE Security Functional Requirements addressed: FCS_IPSEC_EXT.1.7, FCS_IPSEC_EXT.1.8.

7.1.2.4.7 Random Number Generation

The TOE supports DH Group 5, Group 14, 256-bit Random ECP Group (Group 19) and 384-bit Random ECP Group (Group 20). Random numbers are generated using the Cavium Octeon hardware, and from software using input from the devices address, time, RAM, and the configuration buffer contents. The software sources are combined with the hardware source using a SHA-256-bit hash which distributes the entropy over the entire data set. This is used to generate ‘x’, where ‘x’ is the secret value used in the IKE Diffie-Hellman key exchange.

For Group 5 and Group 14 implementations, random bytes are taken from the entropy source with the length of ‘x’. For Group 19 and Group 20 implementations, random bytes are taken from the entropy source with the length of ‘x’, and further examined to ensure that ‘x’ < (order -1), where order is one of the domain parameters of the Elliptic Curve.

The number of random bits used to generate ‘x’ is dependent upon the group as follows:

- Group 5: 256
- Group 14: 320
- 256-bit Random ECP Group (Group 19): 256
- 384-bit Random ECP Group (Group 20): 384

The bits of security value, as detailed in the NDcPP and NIST Special Publication 800-57 Part 1 is as follows:

- Group 5: 90
- Group 14: 112
- 256-bit Random ECP Group (Group 19): 128
- 384-bit Random ECP Group (Group 20): 192

TOE Security Functional Requirements addressed: FCS_IPSEC_EXT.1.9

7.1.2.4.8 Nonce Length

The DRBG described in FCS_RBG_EXT.1 is used to generate each nonce for DH groups 5, 14, 19 and 20 for IKEv1. The output of the DRBG is 20 bytes, which meets the stipulations of the requirement.

TOE Security Functional Requirements addressed: FCS_IPSEC_EXT.1.10.

7.1.2.4.9 DH Group Support

DH Groups 5, 14, 256-bit Random ECP Group (Group 19) and 384-bit Random ECP Group (Group 20) are supported. The group is identified in the policy and

must be the same for both ends of the tunnel in order for negotiations to proceed.

TOE Security Functional Requirements addressed: FCS_IPSEC_EXT.1.11.

7.1.2.4.10 Potential Strength

The symmetric algorithm supported for IKEv1 Phase1 (AES-256) uses the same or greater key length as the symmetric algorithms used to protect IKEv1 Phase2 (AES-128 and AES-256).

The available options ensure that the IKEv1 Phase1 symmetric algorithm key length is equal to or greater than the IKEv1 Phase2 symmetric algorithm key length. Therefore, no checks are required..

TOE Security Functional Requirements addressed: FCS_IPSEC_EXT.1.12.

7.1.2.4.11 Peer Authentication

Peer authentication is performed using third-party RSA certificates.

The third-party certificate must be signed using 2048-bit RSA and installed on the TOE. It is then selected from the Gateway Certificate drop down list. Only 2048-bit RSA certificates may be used in the evaluated configuration. Peer certificates must also be configured to be accepted.

TOE Security Functional Requirements addressed: FCS_IPSEC_EXT.1.13.

7.1.2.4.12 Distinguished Name comparisons

When certificates are used, the Distinguished Name in the certificate is compared with the Distinguished Name presented in the request.

The format of any Subject Distinguished Name is determined by the issuing Certification Authority. Common fields are Country (C=), Organization (O=), Organizational Unit (OU=), Common Name (CN=), Locality (L=), and vary with the issuing Certification Authority. The actual Subject Distinguished Name field in an X.509 Certificate is a binary object which is converted to a string and compared with the expected string.

TOE Security Functional Requirements addressed: FCS_IPSEC_EXT.1.14.

7.1.2.5 Random Bit Generation

The TOE implements a DRBG in accordance with ISO/IEC 18031:2011 using Hash_DRBG. The entropy source is discussed in the Entropy documentation. Entropy was not tested. The min-entropy is assumed to be 0.4 bits of entropy per bit.

TOE Security Functional Requirements addressed: FCS_RBG_EXT.1.

7.1.2.6 TLS Server Protocol

The TLS Server protocol is implemented in support of the HTTPS connection to the administrative interface. The following ciphersuites are supported:

- TLS_RSA_WITH_AES_128_CBC_SHA
- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_RSA_WITH_AES_128_CBC_SHA256
- TLS_RSA_WITH_AES_256_CBC_SHA256

The TOE supports TLS 1.1 and TLS 1.2. All other protocol requests will be denied. RSA with 2048 bit keys and Diffie-Hellman 2048 bit parameters are implemented in these ciphersuites.

Additional detail on the SonicWall implementation of TLS may be found in SSL Control, Chapter 79 of the SonicWall SonicOS v6.2 Administration Guide.

TOE Security Functional Requirements addressed: FCS_HTTPS_EXT.1, FCS_TLSS_EXT.1.

7.1.3 Identification and Authentication

7.1.3.1 User Access and Password Management

The SonicOS Management UI is the application used to manage the TOE devices. It is protected by HTTPS. An HTTPS session is established with the appliance. Then, a login screen displaying the administrator-configured warning banner is presented to users, and the user must be identified and authenticated prior to being granted access to any security functionality. In the evaluated configuration, only the local authentication mechanism (where username and password are stored within the device) is supported.

The logon process requires that the user enter the username and password on the logon screen. Passwords are obscured with dots to prevent an unauthorized individual from inadvertently viewing the password. Passwords must meet the rules set by the administrator. These rules are governed by the requirements described in FIA_PMG_EXT.1.

A user will only be granted access to the SonicOS Management UI Dashboard if authentication is successful. If unsuccessful, the logon screen will be displayed.

TOE Security Functional Requirements addressed: FIA_PMG_EXT.1, FIA_UIA_EXT.1, FIA_UAU_EXT.2, FIA_UAU.7.

7.1.3.2 X.509 Certificate Path Validation

The validity of certificates is checked on certificate import and prior to usage of the public key within the certificate. Certificate validation includes checks of:

- the certificate date
- the validation path, ensuring that the certificate path terminates with a trusted CA certificate
- basicConstraints, ensuring the presence of the basicConstraints extention
- revocation status, using OCSP
- extendedKeyUsage properties, if the certificate is used for trusted updates, client authentication or OCSP

The certificate path validation algorithm is implemented as described in RFC 5280.

The certificate path is also validated when a certificate is imported. This validation includes a check of the certificate chain, and the keys of each of the certificates in the chain. The validity period of the certificate is also checked at this time. If a CRL is imported with the certificate, the CRL is also checked at this time. When the certificate is used, the OCSP server is contacted to verify that the certificate is still valid.

TOE Security Functional Requirements addressed: FIA_X509_EXT.1, FIA_X509_EXT.3.

7.1.3.3 X.509 Certificate Usage

Certificates are used for IPsec, TLS, and HTTPS.

Certificates used for IPsec are assigned a name when imported, and are selected by name when the parameters are selected for an IPsec Security Policy.

The certificate used for TLS/HTTPS is called the 'HTTPS Management Certificate', and is created for that purpose on the TOE device.

If the validity of a certificate cannot be verified, the system rejects the certificate and drops the connection. The TOE may be configured to accept or reject self-signed certificates.

TOE Security Functional Requirements addressed: FIA_X509_EXT.2.

7.1.4 Security Management

The TOE security functions are managed locally and remotely through the web-based management interface and restricted to authorized users assigned the Security Administrator role. Security Administrators must authenticate with the TOE prior to accessing any of the administrative functions. Manual updates to the TOE may only be performed by Security Administrators. No management of TSF data may be performed through any interface prior to login. Only administrators may login to the administrative interface, ensuring that access to TSF data is disallowed for non-administrative users.

TOE Security Functional Requirements addressed: FMT_MOF.1(1)/TrustedUpdate, FMT_MTD.1, FMT_SMF.1, FMT_SMR.2.

7.1.5 Protection of the TSF

7.1.5.1 Protection of Administrator Passwords

The TSF protects the administrator passwords used to access the device. Passwords are passed through a hash function, and only the resulting hash is stored. The user interface does not support viewing of passwords.

TOE Security Functional Requirements addressed: FPT_APW_EXT.1.

7.1.5.2 Protection of TSF Data

The TSF does not include any function that allows symmetric keys or private keys to be displayed or exported. The use of shared secrets is not supported in the evaluated configuration. Keys may only be accessed for the purposes of their assigned security functionality.

Key storage is detailed in Table 14.

TOE Security Functional Requirements addressed: FPT_SKP_EXT.1.

7.1.5.3 Reliable Time Stamps

The TOE provides reliable time stamps that support all TOE functions. The System > Time page of the web management GUI may be used to configure the time and date settings. In the evaluated configuration, time is set manually. This may be configured by deselecting 'Set time automatically using NTP' and populating the appropriate values for daylight savings time adjustments and time format. Only authorized administrators have the required privilege to set the time.

The following security functions make use of the provided time:

- Audit records
- Dashboard displays
- Traffic Statistics
- System Schedules
- Reporting

Time is maintained by the system clock, which is implemented in the TOE hardware and software. Changes to the time are audited. Therefore, the time services provided are considered to be reliable.

Authorized administrators may make changes to the time using the GUI.

TOE Security Functional Requirements addressed: FPT_STM.1.

7.1.5.4 TSF Testing

The TOE performs a power on self-test on each device when it is powered on. The following tests are performed:

- CPU Test - This includes tests and set-up of the following:
  - MMU
  - Memory
  - I/O ports
  - Interrupts
  - Timers
- RAM Test - A memory test is performed.

Following these tests, the TSF performs self-tests on the cryptographic module. The following cryptographic algorithm self-tests are performed by the cryptographic module at power-up:

- Firmware integrity test (using 16-bit CRC EDC)

- AES-CBC Encrypt and Decrypt Known Answer Tests
- SHA-1, -256, -384, -512 Known Answer Tests
- HMAC-SHA-1, -256, -512 Known Answer Tests
- DSA Signature Verification Pairwise Consistency Test
- RSA Sign and Verify Known Answer Tests
- DH Pairwise Consistency Test
- DRBG Known Answer Test

When a new firmware image is loaded, the cryptographic module verifies the ECDSA signed SHA-2 hash of the image. If this verification fails, the firmware image loading is aborted.

If any of the tests fail, the cryptographic module enters the error state. No security services are provided in the error state. Upon successful completion of the Diagnostic Phase, the cryptographic module enters the Command and Traffic Processing State. Security services are only provided in the Command and Traffic Processing State. No VPN tunnels are started until all tests are successfully completed. This effectively inhibits the data output interface. When all tests are completed successfully, the Test Light Emitting Diode (LED) is turned off.

The SonicWall device is essentially a Finite State Machine that is synonymous with the cryptographic module. Therefore, the cryptographic module self-tests are entirely sufficient to demonstrate the correct operation of the TOE.

TOE Security Functional Requirements addressed: FPT_TST_EXT.1.

7.1.5.5 Trusted Update

TSF software may be updated through the web interface using the System > Settings page. This page displays the current firmware image version. To update the firmware, the administrator must first download the firmware update from SonicWall and save it to an accessible location. The administrator then selects the ‘Upload New Firmware’ button and ‘Browse’ to navigate to the firmware on the local drive. Once selected, the administrator selects ‘Upload’. The digital signature on the firmware is automatically verified using the SonicWall public key. This key is appended to each firmware image made available to customers, and is used to verify the new firmware. If the signature verification succeeds, the firmware is automatically installed. If the signature verification fails, the firmware is not uploaded and an error appears.

Newly installed firmware is inactive until being activated by an administrator. The new firmware can be activated through the web interface by navigating to Status > Settings and clicking the “Boot” option located beside the new firmware version.

TOE Security Functional Requirements addressed: FPT_TUD_EXT.1

7.1.6 TOE Access

All access to the TOE takes place through the web-based management interface over HTTPS. The web-based management interface may be accessed using the

GUI (Note that the Getting Started or Quick Start Guide refers to the GUI as the MGMT interface). In the evaluated configuration, no management interfaces are configured to allow use of the Command Line Interface (CLI).

Inactive local and remote sessions to the TOE are automatically terminated after a Security Administrator-configurable time interval between 1 and 9999 minutes. By default, the TOE terminates a session after five minutes of inactivity. In addition, administrators are provided with the capability to terminate their own session. All users are presented with a Security Administrator-configured advisory notice and consent warning at TOE login.

TOE Security Functional Requirements addressed: FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4, FTA_TAB.1.

7.1.7 Trusted Path / Channels

IPsec VPN tunnels are used to provide a trusted communication channel between the TOE and the external audit server. The exchange of information to authenticate the members of the VPN and encrypt/decrypt the data uses the Internet Key Exchange (IKE) protocol for exchanging authentication information (keys) and establishing the VPN tunnel.

HTTPS is used to provide a trusted path for communications between the TOE and the administrative interface. The TOE supports TLS 1.1 and TLS 1.2 to protect these communications from disclosure and detect modification. All other protocol requests will be denied. RSA with 2048 bit keys and Diffie-Hellman 2048 bit parameters are implemented in the supported TLS ciphersuites.

TOE Security Functional Requirements addressed: FTP_ITC.1, FTP_TRP.1.

8 TERMINOLOGY AND ACRONYMS

8.1 TERMINOLOGY

The following terminology is used in this ST:

Term	Description
Administrator	The collaborative Protection Profile for Network Devices (NDcPP) defines the role 'Security Administrator' to administer the TOE. The SonicWall documentation describes a single 'Administrator' role. The terms 'Security Administrator' and 'Administrator' are used interchangeably to describe the user that manages the security functionality of the TOE.
MGMT	The GUI is referred to as the MGMT interface in the TOE guidance.
Security Policy	The term 'security policy' is used in this ST to describe the policies implemented within the TOE to enforce the TOE security features.

Table 17 – Terminology

8.2 ACRONYMS

The following acronyms are used in this ST:

Acronym	Definition
AD	Active Directory
AES	Advanced Encryption Standard
AH	Authentication Header
CA	Certification Authority
CAVP	Cryptographic Algorithm Validation Program
CBC	Cipher Block Chaining
CC	Common Criteria
CLI	Command Line Interface
CM	Configuration Management
CMVP	Cryptographic Module Validation Program
cPP	collaborative Protection Profile
CRC	Cyclical Redundancy Check

Acronym	Definition
CRL	Certificate Revocation List
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard
EAL	Evaluation Assurance Level
ECDSA	Elliptic Curve Digital Signature Algorithm
ECP Group	Elliptic Curve Group modulo a Prime
EDC	Error Detection and Correction
ESP	Encapsulating Security Payload
FFC	Finite Field Cryptography
FIPS	Federal Information Processing Standards
FTP	File Transfer Protocol
GCM	Galois Counter Mode
GUI	Graphical User Interface
HMAC	Hash Message Authentication Code
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
ID	Identification
IKE	Internet Key Exchange
IP	Internet Protocol
IPS	Intrusion Prevention System
IPSec	Internet Protocol Security
ISO/IEC	International Organization for Standardization and the International Electrotechnical Commission
IT	Information Technology
KAT	Known Answer Test
LAN	Local Area Network

Acronym	Definition
LDAP	Lightweight Directory Access Protocol
LED	Light-Emitting Diode
MAC	Message Authentication Code
MODP	Modular Exponential
NAT	Network Address Translation
NDPP	Protection Profile for Network Devices
NDcPP	collaborative Protection Profile for Network Devices
NDRNG	Non-Deterministic Random Number Generator
NIST	US National Institute of Standards and Technology
NTP	Network Time Protocol
OCSP	Online Certificate Status Protocol
OID	Object Identifier
OSP	Organizational Security Policy
PKCS	Public Key Cryptography Standards
PP	Protection Profile
PRF	Pseudorandom Function
RADIUS	Remote Authentication Dial-In User Service
RAM	Random Access Memory
RBG	Random Bit Generator
RFC	Request for Comment
RSA	Rivest, Shamir and Adleman
RSASSA-PSS	RSA Signature Scheme with Appendix - Probabilistic Signature Scheme
SA	Security Association
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SNMP	Secure Network Mail Protocol
SP	Special Publication

Acronym	Definition
SPD	Security Policy Database
SQL	Structured Query Language
SSH	Secure Shell
SSL	Secure Sockets Layer
ST	Security Target
TD	Technical Decision
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality
TSS	TOE Summary Specification
UI	User Interface
URI	Universal Resource Identifier
URL	Universal Resource Locator
VoIP	Voice over Internet Protocol
VPN	Virtual Private Network
WAN	Wide Area Network
WXA	WAN Acceleration

Table 18 – Acronyms