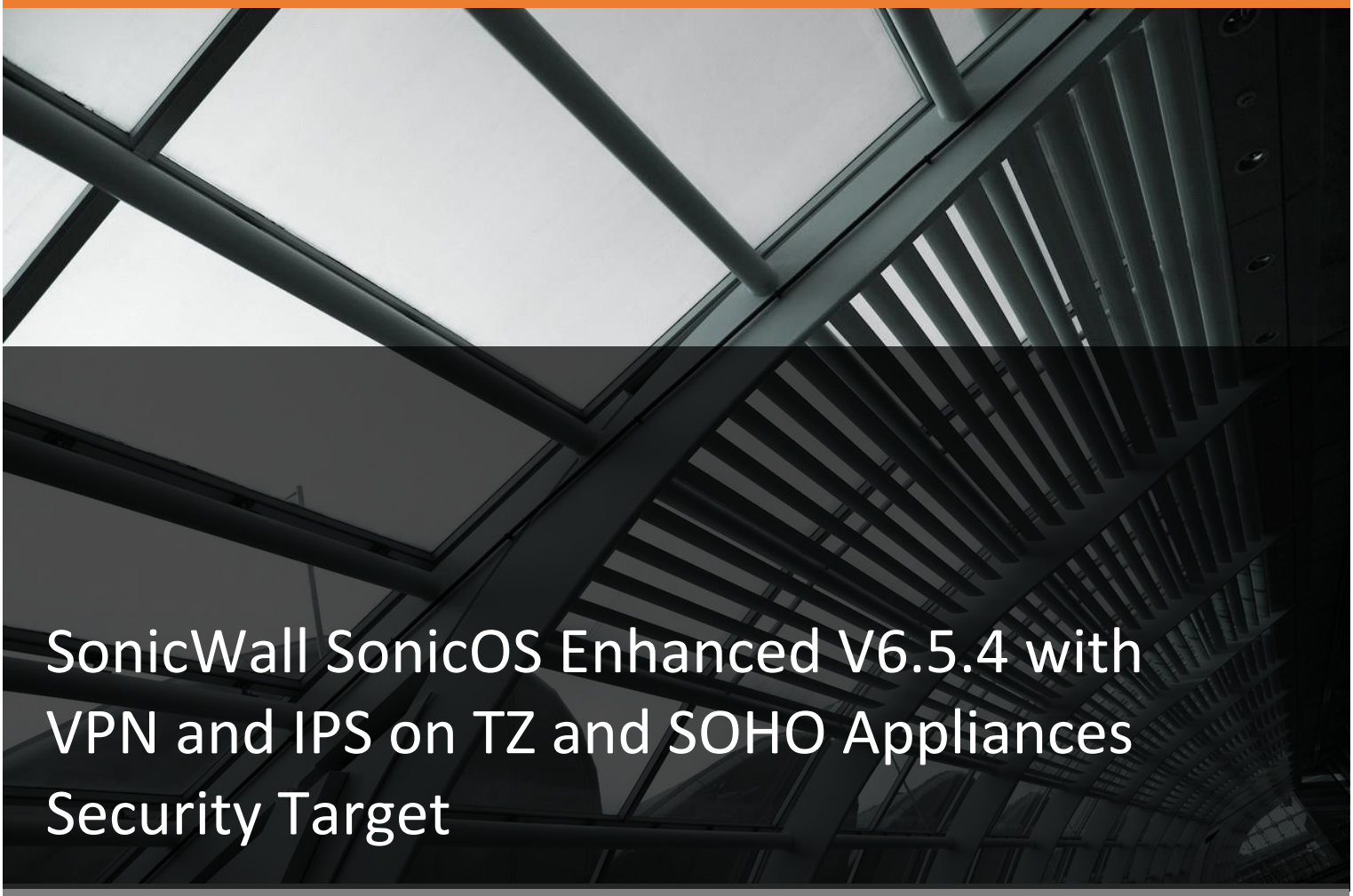

SonicWall SonicOS Enhanced V6.5.4 with VPN and IPS on TZ and SOHO Appliances Security Target

Acumen Security, LLC.

Document Version: 1.9

Table Of Contents

1 Security Target Introduction 6
1.1 Security Target and TOE Reference 6
1.2 TOE Overview 6
1.3 TOE Environment 6
1.4 TOE Architecture 7
1.4.1 Physical Boundaries 7
1.4.2 Security Functions provided by the TOE 7
1.4.2.1 Security Audit 7
1.4.2.2 Cryptographic Support 7
1.4.2.3 Identification and Authentication 8
1.4.2.4 Security Management 9
1.4.2.5 Protection of the TSF 9
1.4.2.6 TOE Access 9
1.4.2.7 Trusted Path/Channels 9
1.4.2.8 Stateful Traffic Filtering 9
1.4.3 TOE Documentation 9
1.5 Functionality Excluded from the Evaluated Configuration 9
2 Conformance Claims 10
2.1 CC Conformance 10
2.2 Protection Profile Conformance 10
2.3 Conformance Rationale 10
2.3.1 Technical Decisions 10
3 Security Problem Definition 16
3.1 Threats 16
3.2 Assumptions 17
3.3 Organizational Security Policies 18
4 Security Objectives 19
4.1 Security Objectives for the Operational Environment 19
5 Security Requirements 20
5.1 Conventions 21
5.2 Security Functional requirements 21
5.2.1 Audit (FAU) 21
5.2.1.1 FAU_GEN.1 Audit data generation 21
5.2.1.2 FAU_GEN.2 User identity association 23
5.2.1.3 FAU_STG_EXT.1 Protected Audit Event Storage 23
5.2.2 Cryptographic Support (FCS) 24
5.2.2.1 FCS_CKM.1 Cryptographic Key Generation 24
5.2.2.2 FCS_CKM.2 Cryptographic Key Establishment 24
5.2.2.3 FCS_CKM.4 Cryptographic Key Destruction 24

5.2.2.4 FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/ Decryption) 24
5.2.2.5 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification). 24
5.2.2.6 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm) 25
5.2.2.7 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm) 25
5.2.2.8 FCS_HTTPS_EXT.1 HTTPS Protocol 25
5.2.2.9 FCS_IPSEC_EXT.1 IPsec Protocol 25
5.2.2.10 FCS_RBG_EXT.1 Random Bit Generation 26
5.2.2.11 FCS_TLSS_EXT.1 TLS Server Protocol 27
5.2.3 User Data Protection (FDP) 27
5.2.3.1 FDP_RIP.2 Full residual information protection 27
5.2.4 Identification and Authentication (FIA) 27
5.2.4.1 FIA_AFL.1 Authentication Failure Heading 27
5.2.4.2 FIA_PMG_EXT.1 Password Management 27
5.2.4.3 FIA_UIA_EXT.1 User identification and authentication 27
5.2.4.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism 28
5.2.4.5 FIA_UAU.7 Protected Authentication Feedback 28
5.2.4.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation 28
5.2.4.7 FIA_X509_EXT.2 X.509 Certificate Authentication 28
5.2.4.8 FIA_X509_EXT.3 X.509 Certificate Requests 28
5.2.5 Security Management (FMT) 28
5.2.5.1 FMT_MOF.1/ManualUpdate Management of security functions behaviour 28
5.2.5.2 FMT_MOF.1/Services Management of security functions behaviour 29
5.2.5.3 FMT_MTD.1/CryptoKeys Management of TSF Data 29
5.2.5.4 FMT_MTD.1/CoreData Management of TSF data 29
5.2.5.5 FMT_SMF.1 Specification of Management Functions 29
5.2.5.6 FMT_SMR.2 Restrictions on Security Roles 29
5.2.6 Protection of TSF (FPT) 29
5.2.6.1 FPT_APW_EXT.1 Protection of administrator passwords 29
5.2.6.2 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) 30
5.2.6.3 FPT_STM_EXT.1 Reliable Time Stamps 30
5.2.6.4 FPT_TST_EXT.1 TSF testing 30
5.2.6.5 FPT_TUD_EXT.1 Trusted update 30
5.2.7 TOE Access (FTA) 30
5.2.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking 30
5.2.7.2 FTA_SSL.3 TSF-initiated Termination 30
5.2.7.3 FTA_SSL.4 User-initiated Termination 30
5.2.7.4 FTA_TAB.1 Default TOE Access Banners 31
5.2.8 Trusted Path/Channel (FTP) 31
5.2.8.1 FTP_ITC.1 Inter-TSF trusted channel 31
5.2.8.2 FTP_TRP.1/Admin Trusted path 31

5.2.9 Stateful Traffic Filter Firewall (FFW) 31
5.2.9.1 FFW_RUL_EXT.1 Stateful traffic filtering 31
5.3 TOE SFR Dependencies Rationale for SFRs 33
5.4 Security Assurance Requirements 33
5.5 Rationale for Security Assurance Requirements 33
5.6 Assurance Measures 34
6 TOE Summary Specification 35

Revision History

Version	Date	Description
1.0	April 2019	Initial Release
1.1	July 2019	Updated with Hypervisor information and TDs
1.2	August 2019	Updated based on internal review.
1.3	September 2019	Updated to adjust TOE scope
1.4	October 2019	Removed headend claims.
1.5	November 2019	Incorporated Technical Decisions
1.6	March 2020	Finalized for submission.
1.7	April 2020	Updated to address ECR comments
1.8	April 2020	Finalized for publication
1.9	April 2020	Updated exclusion list

1 Security Target Introduction

1.1 Security Target and TOE Reference

This section provides information needed to identify and control this ST and its TOE.

Category	Identifier
ST Title	SonicWall SonicOS Enhanced V6.5.4 with VPN and IPS on TZ and SOHO Appliances Security Target
ST Version	1.9
ST Date	April 2020
ST Author	Acumen Security, LLC.
TOE Identifier	SonicWall SonicOS Enhanced V6.5.4 with VPN and IPS on TZ and SOHO Appliances
TOE Software Version	6.5.4.4-44n-federal-12n
TOE Developer	SonicWall, Inc.
Key Words	Firewall

Table 1 TOE/ST Identification

1.2 TOE Overview

The TOE is comprised of the SonicWall SonicOS Enhanced v6.5.4 software running either on purpose built TZ and SOHO hardware appliance platforms.

The appliance firewall capabilities include stateful packet inspection. Stateful packet inspection maintains the state of network connections, such as Transmission Control Protocol (TCP) streams and User Datagram Protocol (UDP) communication, traveling across the firewall. The firewall distinguishes between legitimate packets and illegitimate packets for the given network deployment. Only packets adhering to the administrator-configured access rules are permitted to pass through the firewall; all others are rejected.

The appliance capabilities include deep-packet inspection (DPI) used for intrusion prevention and detection. These services employ stream-based analysis wherein traffic traversing the product is parsed and interpreted so that its content might be matched against a set of signatures to determine the acceptability of the traffic. Only traffic adhering to the administrator-configured policies is permitted to pass through the TOE.

The 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 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 buttons. Administrators may configure policies and manage network traffic, users, and system logs. The appliances also have local console access where limited administrative functionality to configure the network, perform system updates, and view logs.

1.3 TOE Environment

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

1. Management Console - Any computer that provides a supported browser may be used to access the GUI.

2. An audit server supporting the syslog protocol with an IPsec peer supporting IKEv2 and ESP in the cryptographic protocols defined in section 5.2.2.9 of this document.

1.4 TOE Architecture

1.4.1 Physical Boundaries

The TOE is a software and hardware TOE. It is a combination of a particular SOHO or TZ hardware appliance and the SonicOS v6.5.4.4-44n-federal-12n software. The following table 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	Hardware Model	Operational Environment
TZ	TZ 300P	Cavium Octeon III CN7020-800
	TZ 350W	Cavium Octeon III CN7020-800
	TZ 600P	Cavium Octeon III CN7130-1400
SOHO	SOHO 250	Cavium Octeon III CN7020-800
	SOHO 250W	Cavium Octeon III CN7020-800

Table 2 TOE Appliance Series and Models

The underlying platform that comprises the TOE has common hardware characteristics. These differing characteristics effect only non-TSF relevant functionality, such as throughput, processing speed, number and type of connections, and amount of internal storage.

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

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.

1.4.2 Security Functions provided by the TOE

The TOE provides the security functionality required by [FWcPP].

1.4.2.1 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.

1.4.2.2 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 to support Internet Protocol Security (IPsec) to provide VPN functionality and to protect the connection to the audit server.

Algorithm	Description	Mode Supported	CAVP Cert. #
AES	Used for symmetric encryption/decryption FCS_TLSS_EXT.1 FCS_IPSEC_EXT.1 FCS_COP.1/DataEncryption	CBC (128, 256) GCM (128, 256)	C743

Algorithm	Description	Mode Supported	CAVP Cert. #
SHS	Cryptographic hashing services FCS_TLSS_EXT.1 FCS_IPSEC_EXT.1 FCS_RBG_EXT.1 FCS_COP.1/SigGen FCS_COP.1/Hash FCS_COP.1/KeyedHash	SHA (1, 256, 384, 512)	C743
DRBG	Deterministic random bit generation FCS_TLSS_EXT.1 FCS_IPSEC_EXT.1 FCS_RBG_EXT.1 FCS_CKM.1	Hash (SHA-256)	C743
ECDSA (186)	Key Generation, SigGen, SigVer FCS_IPSEC_EXT.1 FCS_CKM.1 FCS_COP.1/SigGen FPT_TUD_EXT.1	P-256, P-384	C743
RSA (186)	Key Generation FCS_TLSS_EXT.1 FCS_IPSEC_EXT.1 FCS_CKM.1	n (2048)	C743
	SigGen (PKCS1_V1.5) FCS_TLSS_EXT.1 FCS_IPSEC_EXT.1 FCS_COP.1/SigGen	n = 2048 SHA(256, 384, 512)	C743
	SigVer (PKCS1_v1.5) FCS_TLSS_EXT.1 FCS_IPSEC_EXT.1 FCS_COP.1/SigGen	n = 2048 SHA(1, 256, 384, 512)	C743
HMAC	Keyed hashing services FCS_TLSS_EXT.1 FCS_IPSEC_EXT.1 FCS_COP.1/KeyedHash	SHA (1, 256, 384, 512)	C743
KAS ECC	SP 800-56A FCS_IPSEC_EXT.1 FCS_CKM.2	Key Agreement (Initiator, Responder) EC: P-256, SHA-512 ED: P-384, SHA-512	C743
RSA	PKCS1_v1.5 FCS_TLSS_EXT.1 FCS_CKM.2	RSA Key Establishment	Vendor Affirmed

Table 3 CAVP Certificate References

1.4.2.3 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.

1.4.2.4 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, manage users and configure the Virtual Private Network (VPN) functionality.

1.4.2.5 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 shuts down if a critical failure occurs. The TOE verifies the software image when it is loaded. The TOE ensures that updates to the TOE software can be verified using a digital signature.

1.4.2.6 TOE Access

The TOE monitors local and remote administrative 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.

1.4.2.7 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.

1.4.2.8 Stateful Traffic Filtering

The TOE restricts the flow of network traffic between protected networks and other attached networks based on addresses and ports of the network nodes originating (source) and/or receiving (destination) applicable network traffic, as well as on established connection information.

1.4.3 TOE Documentation

- SonicWall® SonicOS 6.5 Common Criteria Addendum, Version 1.4

1.5 Functionality Excluded from the Evaluated Configuration

The following features/functionality are excluded from this evaluation:

- Although SonicWall SonicOS Enhanced 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))
- 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
- Intrusion Prevention System (IPS)
- VPN Gateway – Note: IPsec functionality for securing TOE traffic is in scope.

2 Conformance Claims

2.1 CC Conformance

This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 4, September 2012
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 4, September 2012: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 4, September 2012: Part 3 conformant

2.2 Protection Profile Conformance

The TOE for this ST claims exact conformance to the collaborative Protection Profile for Stateful Traffic Filter Firewalls (v2.0+Errata 20180314, 14-March-2018) [FWcPP].

2.3 Conformance Rationale

The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profile performing only operations defined there.

2.3.1 Technical Decisions

The following Technical Decisions have been considered for this evaluation:

TD	REFERENCE	Applicable	Exclusion Rationale
TD0484: NIT Technical Decision for Interactive sessions in FTA_SSL_EXT.1 & FTA_	CPP_FW_V2.0E	Yes	
TD0483: NIT Technical Decision for Applicability of FPT_APW_EXT.1	CPP_FW_V2.0E	Yes	
TD0482: NIT Technical Decision for Identification of usage of cryptographic schemes	CPP_FW_V2.0E	Yes	
TD0481: NIT Technical Decision for FCS_(D)TLSC_EXT.X.2 IP addresses in reference identifiers	CPP_FW_V2.0E	No	FCS_TLSC_EXT.x.2 functionality is not included in this TOE.
TD0480: NIT Technical Decision for Granularity of audit events	CPP_ND_V2.0E	Yes	
TD0478: NIT Technical Decision for Application Notes for FIA_X509_EXT.1 iterations	CPP_FW_V2.0E	Yes	
TD0477: NIT Technical Decision for Clarifying FPT_TUD_EXT.1 Trusted Update	CPP_ND_V2.0E	Yes	
TD0476: NIT Technical Decision for Conflicting FW rules cannot be configured	CPP_FW_V2.0E	Yes	
TD0475: NIT Technical Decision for Separate traffic consideration for SSH rekey	CPP_FW_V2.0E	No	FCS_SSH*_EXT functionality is not included in this TOE.

TD	REFERENCE	Applicable	Exclusion Rationale
TD0453: NIT Technical Decision for Clarify authentication methods SSH clients can use to authenticate SSH se	CPP_FW_V2.0E	No	FCS_SSHC_EXT.1 functionality is not included in this TOE.
TD0451: NIT Technical Decision for ITT Comm UUID Reference Identifier	CPP_FW_V2.0E	No	This TD does not change the requirements.
TD0447: NIT Technical Decision for Using 'diffie-hellman-group-exchange-sha256' in FCS_SSHC/S_EXT.1.7	CPP_FW_V2.0E	No	This TD does not change the requirements.
TD0425: NIT Technical Decision for Cut-and-paste Error for Guidance AA	CPP_ND_V2.0E	Yes	
TD0423: NIT Technical Decision for Clarification about application of RfI#201726rev2	CPP_FW_V2.0E	Yes	
TD0412: NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy	CPP_FW_V2.0E	No	FCS_SSH*_EXT functionality is not included in this TOE.
TD0411: NIT Technical Decisions for FCS_SSHC_EXT.1.5, Test 1 – Server and client side seem to be confused	CPP_FW_V2.0E	No	FCS_SSH*_EXT functionality is not included in this TOE.
TD0410: NIT technical decision for Redundant assurance activities associated with FAU_GEN.1	CPP_ND_V2.0E	Yes	
TD0409: NIT decision for Applicability of FIA_AFL.1 to key-based SSH authentication	CPP_ND_V2.0E	Yes	
TD0408: NIT Technical Decision for Local vs Remote administrator accounts	CPP_FW_V2.0E	Yes	
TD0402: NIT Technical Decision for RSA-based FCS_CKM.2 Selection	CPP_FW_V2.0E	Yes	
TD0400: NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment	CPP_FW_V2.0E	Yes	
TD0399: NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2)	CPP_ND_V2.0E	Yes	
TD0398: NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR	CPP_FW_V2.0E	No	FCS_SSH*_EXT functionality is not included in this TOE.
TD0397: NIT Technical Decision for Fixing AES-CTR Mode Tests	CPP_ND_V2.0E	Yes	
TD0396: NIT Technical Decision for FCS_TLSC_EXT.1.1, Test 2	CPP_ND_V2.0E	No	FCS_TLSC_EXT.1 functionality is not included in this TOE.

TD	REFERENCE	Applicable	Exclusion Rationale
TD0395: NIT Technical Decision for Different Handling of TLS1.1 and TLS1.2	CPP_ND_V2.0E	Yes	
TD0394: NIT Technical Decision for Audit of Management Activities related to Cryptographic Keys	CPP_FW_V2.0E	Yes	
TD0343: NIT Technical Decision for Updating FCS_IPSEC_EXT.1.14 Tests	ND SD V2.0, FCS_IPSEC_EXT.1.14, CPP_FW_V2.0E, CPP_ND_V2.0E	Yes	
TD0342: NIT Technical Decision for TLS and DTLS Server Tests	ND SD V2.0, FCS_DTLSS_EXT.1, FCS_DTLSS_EXT.2, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2, CPP_ND_V2.0E	Yes	
TD0341: NIT Technical Decision for TLS wildcard checking	ND SD V2.0, FCS_TLSC_EXT.1.2, FCS_TLSC_EXT.2.2, FCS_DTLSC_EXT.1.2, FCS_DTLSC_EXT.2.2, CPP_ND_V2.0E	No	FCS_[D]TLSC_EXT.[1|2] functionality is not included in this TOE.
TD0340: NIT Technical Decision for Handling of the basicConstraints extension in CA and leaf certificates	FIA_X509_EXT.1.1, CPP_FW_V2.0E, CPP_ND_V2.0E	Yes	
TD0339: NIT Technical Decision for Making password-based authentication optional in FCS_SSHS_EXT.1.2	ND SD V2.0, FCS_SSHS_EXT.1.2, CPP_FW_V2.0E, CPP_ND_V2.0E	No	FCS_SSHS_EXT.1 functionality is not included in this TOE.
TD0338: NIT Technical Decision for Access Banner Verification	ND SD V2.0, FTA_TAB.1, CPP_ND_V2.0E	Yes	
TD0337: NIT Technical Decision for Selections in FCS_SSH*_EXT.1.6	ND SD V2.0, FCS_SSHC_EXT.1, FCS_SSHS_EXT.1, CPP_FW_V2.0E, CPP_ND_V2.0E	No	FCS_SSH[C|S]_EXT.1 functionality is not included in this TOE.
TD0336: NIT Technical Decision for Audit requirements for FCS_SSH*_EXT.1.8	ND SD V2.0, FCS_SSHC_EXT.1.8, FCS_SSHS_EXT.1.8, CPP_ND_V2.0E	No	FCS_SSH[C|S]_EXT.1 functionality is not included in this TOE.

TD	REFERENCE	Applicable	Exclusion Rationale
TD0335: NIT Technical Decision for FCS_DTLS Mandatory Cipher Suites	FCS_DTLSC_EXT.1.1, FCS_DTLSC_EXT.2.1, FCS_DTLSS_EXT.1.1, FCS_DTLSS_EXT.2.1, FCS_TLSC_EXT.1.1, FCS_TLSC_EXT.2.1, FCS_TLSS_EXT.1.1, FCS_TLSS_EXT.2.1, CPP_FW_V2.0E, CPP_ND_V2.0E	Yes	
TD0334: NIT Technical Decision for Testing SSH when password-based authentication is not supported	ND SD V2.0, FCS_SSHC_EXT.1.9, CPP_ND_V2.0E	No	FCS_SSHC_EXT.1 functionality is not included in this TOE.
TD0333: NIT Technical Decision for Applicability of FIA_X509_EXT.3	ND SD V2.0, FIA_X509_EXT, CPP_FW_V2.0E, CPP_ND_V2.0E	Yes	
TD0324: NIT Technical Decision for Correction of section numbers in SD Table 1	Table 1, CPP_ND_V2.0E	Yes	
TD0323: NIT Technical Decision for DTLS server testing - Empty Certificate Authorities list	ND SD V2.0, FCS_DTLSS_EXT.2.7, FCS_DTLSS_EXT.2.8, CPP_ND_V2.0E	No	FCS_DTLSS_EXT.2 functionality is not included in this TOE.
TD0322: NIT Technical Decision for TLS server testing - Empty Certificate Authorities list	ND SD V.1.0, ND SD V2.0, FCS_TLSS_EXT.2.4, FCS_TLSS_EXT.2.5, CPP_ND_V2.0E	No	FCS_TLSS_EXT.2 functionality is not included in this TOE.
TD0321: Protection of NTP communications	FTP_ITC.1, FPT_STM_EXT.1, CPP_FW_V2.0E, CPP_ND_V2.0E	Yes	
TD0291: NIT Technical Decision for DH14 and FCS_CKM.1	FCS_CKM.1 CPP_FW_V1.0, CPP_FW_v2.0, CPP_FW_V2.0E, CPP_ND_V1.0, CPP_ND_V2.0, CPP_ND_V2.0E, ND SD V.1.0, ND SD V2.0	Yes	
TD0290: NIT Technical Decision for physical interruption of trusted path/channel	FTP_ITC.1, FTP_TRP.1, FPT_ITT.1 CPP_ND_V1.0, CPP_ND_V2.0, CPP_ND_V2.0E, ND SD V.1.0, ND SD V2.0	Yes	

TD	REFERENCE	Applicable	Exclusion Rationale
TD0289: NIT Technical Decision for FCS_TLSC_EXT.x.1 Test 5e	CPP_ND_V1.0, CPP_ND_V2.0, CPP_ND_V2.0E FCS_TLSC_EXT.1.1, FCS_TLSC_EXT.2.1, FCS_DTLSC_EXT.1.1 (only ND SD V2.0) , FCS_DTLSC_EXT.2.1 (only ND SD V2.0)	No	FCS_[D]TLSC_EXT.[1|2] functionality is not included in this TOE.
TD0281 : NIT Technical Decision for Testing both thresholds for SSH rekey	FCS_SSHC_EXT.1.8, FCS_SSHS_EXT.1.8, CPP_ND_V1.0, CPP_ND_V2.0, CPP_ND_V2.0E ND SD V1.0, ND SD V2.0	No	FCS_SSH[C|S]_EXT.1 functionality is not included in this TOE.
TD0259: NIT Technical Decision for Support for X509 ssh rsa authentication IAW RFC 6187	CPP_FW_v2.0, CPP_FW_V2.0E, CPP_ND_V2.0, CPP_ND_V2.0E FCS_SSHC_EXT.1.5/FCS_SSHS_EXT.1.5	No	FCS_SSH[C|S]_EXT.1 functionality is not included in this TOE.
TD0257: NIT Technical Decision for Updating FCS_DTLSC_EXT.x.2/FCS_TLSC_EXT.x.2 Tests 1-4	ND SD V1.0, ND SD V2.0, FCS_DTLSC_EXT.1.2/FCS_DTLSC_EXT.2.2 Tests 1-4 (ND SD V2.0), FCS_TLSC_EXT.1.2/FCS_TLSC_EXT.2.2, Tests 1-4 (ND SD V1.0, ND SD V2.0) CPP_ND_V1.0, CPP_ND_V2.0, CPP_ND_V2.0E	No	FCS_[D]TLSC_EXT.[1|2] functionality is not included in this TOE.
TD0256: NIT Technical Decision for Handling of TLS connections with and without mutual authentication	ND SD V1.0, ND SD V2.0, FCS_DTLSC_EXT.2.5 (ND SD V2.0), FCS_TLSC_EXT.2 (ND SD V1.0, ND SD V2.0) CPP_ND_V1.0, CPP_ND_V2.0, CPP_ND_V2.0E	No	FCS_[D]TLSC_EXT.2 functionality is not included in this TOE.
TD0228: NIT Technical Decision for CA certificates - basicConstraints validation	ND SD V1.0, ND SD V2.0, FIA_X509_EXT.1.2 CPP_FW_V1.0, CPP_ND_V1.0, CPP_ND_V2.0, CPP_ND_V2.0E	Yes	

3 Security Problem Definition

The security problem definition has been taken from [FWcPP] and is reproduced here for the convenience of the reader. The security problem is described in terms of the threats that the TOE is expected to address, assumptions about the operational environment, and any organizational security policies that the TOE is expected to enforce.

Since this TOE is not a distributed TOE, items that only apply to distributed TOEs are not included.

3.1 Threats

ID	Threat
T.UNAUTHORIZED_ ADMINISTRATOR_ACCESS	Threat agents may attempt to gain administrator access to the firewall by nefarious means such as masquerading as an administrator to the firewall, masquerading as the firewall 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 the firewall and a network device. Successfully gaining administrator access allows malicious actions that compromise the security functionality of the firewall 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 firewalls that do not use standardized secure tunnelling 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 firewall itself.
T.WEAK_AUTHENTICATION _ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a 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 and integrity, and potentially the firewall 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 firewall without Administrator awareness. This could

ID	Threat
	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 firewall data enabling continued access to the firewall and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or firewall 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 firewall. Having privileged access to the firewall 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	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.
T.NETWORK_DISCLOSURE	An attacker may attempt to "map" a subnet to determine the machines that reside on the network, and obtaining the IP addresses of machines, as well as the services (ports) those machines are offering. This information could be used to mount attacks to those machines via the services that are exported.
T. NETWORK_ACCESS	With knowledge of the services that are exported by machines on a subnet, an attacker may attempt to exploit those services by mounting attacks against those services.
T.NETWORK_MISUSE	An attacker may attempt to use services that are exported by machines in a way that is unintended by a site's security policies. For example, an attacker might be able to use a service to "anonymize" the attacker's machine as they mount attacks against others.
T.MALICIOUS_TRAFFIC	An attacker may attempt to send malformed packets to a machine in hopes of causing the network stack or services listening on UDP/TCP ports of the target machine to crash.

Table 4 Threats

3.2 Assumptions

ID	Assumption
A.PHYSICAL_PROTECTION	The firewall 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 firewall's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the firewall 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 firewall that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the firewall.

A.LIMITED_FUNCTIONALITY	The firewall 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 firewall device should not provide a computing platform for general purpose applications (unrelated to networking/filtering functionality).
A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the firewall 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 firewall. The firewall 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 firewall 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 firewall device are protected by the platform on which they reside.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on firewall equipment when the equipment is discarded or removed from its operational environment.

Table 5 Assumptions

3.3 Organizational Security Policies

ID	Assumption
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 OSPs

4 Security Objectives

The security objectives have been taken from [FWcPP] and are reproduced here for the convenience of the reader.

Since this TOE is not a distributed TOE, items that only apply to distributed TOEs are not included.

4.1 Security Objectives for the Operational Environment

The following security objectives for the operational environment assist the TOE in correctly providing its security functionality. These track with the assumptions about the environment.

ID	Objective for the Operation Environment
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.TRUSTED_ADMIN	Security 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.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

Table 7 Objectives for the environment

5 Security Requirements

This section identifies the Security Functional Requirements for the TOE and/or Platform. The Security Functional Requirements included in this section are derived from Part 2 of the Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, dated: September 2012 and all international interpretations.

Requirement	Requirement Description
FAU_GEN.1	Audit data generation
FAU_GEN.2	User identity association
FAU_STG_EXT.1	Protected audit event storage
FCS_CKM.1	Cryptographic key generation
FCS_CKM.2	Cryptographic key establishment
FCS_CKM.4	Cryptographic key Destruction
FCS_COP.1/DataEncryption	Cryptographic operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash	Cryptographic operation (Hash algorithm)
FCS_COP.1/KeyedHash	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
FDP_RIP.2	Full residual information protection
FIA_AFL.1	Authentication Failure Handling
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/Rev	X.509 certificate validation
FIA_X509_EXT.2	X.509 certificate authentication
FIA_X509_EXT.3	X.509 certificate requests
FMT_MOF.1/ManualUpdate	Management of security functions behaviour
FMT_MOF.1/Services	Management of security functions behaviour
FMT_MTD.1/CryptoKeys	Management of TSF data
FMT_MTD.1/CoreData	Management of TSF data
FMT_SMF.1	Specification of management functions
FMT_SMR.2	Restrictions on security roles
FPT_APW_EXT.1	Protection of administrator passwords
FPT_SKP_EXT.1	Protection of TSF data (for reading of all symmetric keys)
FPT_STM_EXT.1	Reliable time stamps
FPT_TST_EXT.1	TSF testing
FPT_TUD_EXT.1	Trusted update
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
FTP_ITC.1	Inter-TSF trusted channel
FTP_TRP.1/Admin	Trusted path
FFW_RUL_EXT.1	Stateful traffic filtering

Table 8 SFRs

5.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with *italicized* text;
- Refinement: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending a slash followed by a short description, e.g., /SigGen, /IKE.
- Where operations were completed in the PP itself, the formatting used in the PP has been retained.

Extended SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs. Formatting conventions outside of operations matches the formatting specified within the PP.

5.2 Security Functional requirements

5.2.1 Audit (FAU)

5.2.1.1 FAU_GEN.1 Audit data generation

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

a) Start-up and shut-down 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).*
- *Changes to TSF data related to 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]];*

d) *Specifically defined auditable events listed in Table 9*.

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of Table 9*.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	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
FDP_RIP.2	None.	None.
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of 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/Rev	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Services	Starting and stopping of services.	None.
FMT_MTD.1/CryptoKeys	Management of cryptographic keys.	None.
FMT_MTD.1/CoreData	All management activities of TSF data.	None.
FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous 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).
FPT_TST_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FTA_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session locking mechanism.	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.
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/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.
FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface
	Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets Identifier of rule causing packet drop

Table 9 Security Functional Requirements and Auditable Events

5.2.1.2 FAU_GEN.2 User identity association

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.

5.2.1.3 FAU_STG_EXT.1 Protected Audit Event Storage

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.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 FCS_CKM.1 Cryptographic Key Generation

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;*
- *ECC schemes using "NIST curves" [P-256, P-384] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;*
- *FFC Schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3*

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

5.2.2.2 FCS_CKM.2 Cryptographic Key Establishment

FCS_CKM.2.1 The TSF shall **perform** cryptographic **key establishment** in accordance with a specified cryptographic key **establishment** method: *[*

- *RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*
- *Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";*
- *Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3;*

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

5.2.2.3 FCS_CKM.4 Cryptographic Key Destruction

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

that meets the following: *No Standard*.

5.2.2.4 FCS_COP.1/DataEncryption Cryptographic Operation (AES Data Encryption/ Decryption)

FCS_COP.1.1/DataEncryption The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in [GCM, CBC] mode* and cryptographic key sizes *[128 bits, 256 bits]* that meet the following: *AES as specified in ISO 18033-3, [CBC as specified in ISO 10116, GCM as specified in ISO 19772]*.

5.2.2.5 FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)

FCS_COP.1.1/SigGen 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],*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256-bits, 384-bits]*

] ~~and cryptographic key sizes [assignment: cryptographic key sizes]~~

that meet 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,*
- *For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384]; ISO/IEC 14888-3, Section 6.4*

].

5.2.2.6 FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

FCS_COP.1.1/Hash 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*~~] and **message digest sizes [*160, 256, 384, 512*] bits** that meet the following: *ISO/IEC 10118-3:2004*.

5.2.2.7 FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)

FCS_COP.1.1/KeyedHash The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm *[HMAC-SHA-1, HMAC-SHA256, HMAC-SHA-384, HMAC-SHA-512]* and cryptographic key sizes *[160, 256, 384, 512]* **and message digest sizes [*160, 256, 384, 512*] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

5.2.2.8 FCS_HTTPS_EXT.1 HTTPS 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 If a peer certificate is presented, the TSF shall [*not require client authentication*] if the peer certificate is deemed invalid.

5.2.2.9 FCS_IPSEC_EXT.1 IPsec Protocol

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 [*tunnel 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-CBC256 (specified in RFC 3602)*] together with a Secure Hash Algorithm (SHA)-based HMAC [*HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512*] and [*AES-GCM-128, AES-GCM-256 (specified in RFC 4106)*].

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [

- *IKEv2 as defined in RFC 5996 and [with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)], and [RFC 4868 for hash functions]*

].

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [*IKEv2*] protocol uses the cryptographic algorithms [*AES-CBC-128, AES-CBC-256 (specified in RFC 3602)*].

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [

- *IKEv2 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 24] hours]*

].

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [

- *IKEv2 Child 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 [*224, 256, 384*] bits.

FCS_IPSEC_EXT.1.10 The TSF shall generate nonces used in [*IKEv2*] exchanges of length [

- *[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 Group(s) [*14 (2048-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 [*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 [*IKEv2 CHILD_SA*] connection.

FCS_IPSEC_EXT.1.13 The TSF shall ensure that all IKE protocols perform peer authentication using [*RSA, ECDSA*] 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 if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: [SAN: IP address, SAN: Fully Qualified Domain Name (FQDN), SAN: user FQDN, Distinguished Name (DN)] and [*no other reference identifier type*].

5.2.2.10 FCS_RBG_EXT.1 Random Bit Generation

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 [*[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.

5.2.2.11 FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1 The TSF shall implement [*TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

[

- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *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 2.0, SSL 3.0, TLS 1.0 and [*none*].

FCS_TLSS_EXT.1.3 The TSF shall [*perform RSA key establishment with key size [2048 bits]*].

5.2.3 User Data Protection (FDP)

5.2.3.1 FDP_RIP.2 Full residual information protection

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [*deallocation of the resource from*] all objects.

5.2.4 Identification and Authentication (FIA)

5.2.4.1 FIA_AFL.1 Authentication Failure Heading

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1-99*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall [*prevent the offending Administrator from successfully establishing remote session using any authentication method that involves a password until an Administrator defined time period has elapsed*].

5.2.4.2 FIA_PMG_EXT.1 Password Management

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: [*"!", "@", "#", "$", "%", "^", "&", "*", "(", ")"*];

b) Minimum password length shall be configurable to [*1 character*] and [*99 characters*].

5.2.4.3 FIA_UIA_EXT.1 User identification and authentication

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.

5.2.4.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local [*password-based*] authentication mechanism to perform local administrative user authentication.

5.2.4.5 FIA_UAU.7 Protected Authentication Feedback

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

5.2.4.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

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

- RFC 5280 certificate validation and certificate path validation **supporting a minimum path length of three certificates**.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 6960]
- 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/Rev 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.

5.2.4.7 FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [IPsec, TLS], *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*].

5.2.4.8 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1 The TSF shall generate a Certification Request 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.

5.2.5 Security Management (FMT)

5.2.5.1 FMT_MOF.1/ManualUpdate Management of security functions behaviour

FMT_MOF.1.1/ManualUpdate The TSF shall restrict the ability to enable the functions *to perform manual updates to Security Administrators*.

5.2.5.2 FMT_MOF.1/Services Management of security functions behaviour

FMT_MOF.1.1/Services The TSF shall restrict the ability to enable and disable the functions **and services** to *Security Administrators*.

5.2.5.3 FMT_MTD.1/CryptoKeys Management of TSF Data

FMT_MTD.1.1/CryptoKeys The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

5.2.5.4 FMT_MTD.1/CoreData Management of TSF data

FMT_MTD.1.1/CoreData The TSF shall restrict the ability to *manage* the *TSF data* to *Security Administrators*.

5.2.5.5 FMT_SMF.1 Specification of Management Functions

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;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- *Ability to configure firewall rules;*
- *[*
    - *Ability to configure the cryptographic functionality;*
    - *Ability to configure the lifetime for IPsec SAs;*
    - *Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1;*
    - *Ability to set the time which is used for time-stamps;*

  ]

].

5.2.5.6 FMT_SMR.2 Restrictions on Security Roles

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.

5.2.6 Protection of TSF (FPT)

5.2.6.1 FPT_APW_EXT.1 Protection of administrator passwords

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

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

5.2.6.2 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

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

5.2.6.3 FPT_STM_EXT.1 Reliable Time Stamps

FPT_STM_EXT.1.1 The TSF shall be able to provide reliable time stamps for its own use.

FPT_STM_EXT.1.2 The TSF shall [*allow the Security Administrator to set the time*].

5.2.6.4 FPT_TST_EXT.1 TSF testing

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*
- *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*
- *ECDSA Known Answer Test*
- *ECSDA Signature and Verification Known Answer Tests*

].

5.2.6.5 FPT_TUD_EXT.1 Trusted update

FPT_TUD_EXT.1.1 The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software and [*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.

5.2.7 TOE Access (FTA)

5.2.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

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

- *terminate the session*]

after a Security Administrator-specified time period of inactivity.

5.2.7.2 FTA_SSL.3 TSF-initiated Termination

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

5.2.7.3 FTA_SSL.4 User-initiated Termination

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

5.2.7.4 FTA_TAB.1 Default TOE Access Banners

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

5.2.8 Trusted Path/Channel (FTP)

5.2.8.1 FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall **be capable of using [*IPsec*] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [*[VPN communications]*]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from **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*].

5.2.8.2 FTP_TRP.1/Admin Trusted path

FTP_TRP.1.1/Admin The TSF shall **be capable of using [*TLS, HTTPS*] to** provide a communication path between itself and **authorized** remote **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/Admin The TSF shall permit remote **Administrators** to initiate communication via the trusted path.

FTP_TRP.1.3/Admin The TSF shall require the use of the trusted path for *initial Administrator authentication and all remote administration actions*.

5.2.9 Stateful Traffic Filter Firewall (FFW)

5.2.9.1 FFW_RUL_EXT.1 Stateful traffic filtering

FFW_RUL_EXT.1.1 The TSF shall perform Stateful Traffic Filtering on network packets processed by the TOE.

FFW_RUL_EXT.1.2 The TSF shall allow the definition of Stateful Traffic Filtering rules using the following network protocol fields:

- ICMPv4
  - Type
  - Code
- ICMPv6
  - Type
  - Code
- IPv4
  - Source address
  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - Transport Layer Protocol

    - [*no other field*]
- TCP
    - Source Port
    - Destination Port
- UDP
    - Source Port
    - Destination Port
- and distinct interface.

FFW_RUL_EXT.1.3 The TSF shall allow the following operations to be associated with Stateful Traffic Filtering rules: permit or drop with the capability to log the operation.

FFW_RUL_EXT.1.4 The TSF shall allow the Stateful Traffic Filtering rules to be assigned to each distinct network interface.

FFW_RUL_EXT.1.5 The TSF shall:

a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [*no other protocols*] based on the following network packet attributes:
    1. TCP: source and destination addresses, source and destination ports, sequence number, Flags;
    2. UDP: source and destination addresses, source and destination ports;
    3. [*no other protocols*].
b) Remove existing traffic flows from the set of established traffic flows based on the following: [*session inactivity timeout, completion of the expected information flow*].

FFW_RUL_EXT.1.6 The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic:

a) The TSF shall drop and be capable of [*logging*] packets which are invalid fragments;
b) The TSF shall drop and be capable of [*logging*] fragmented packets which cannot be re-assembled completely;
c) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a broadcast network;
d) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a multicast network; The TSF shall drop and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
e) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address “reserved for future use” (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
f) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as an “unspecified address” or an address “reserved for future definition and use” (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;
g) The TSF shall drop and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
h) [*no other rules*].

FFW_RUL_EXT.1.7 The TSF shall be capable of dropping and logging according to the following rules:

a) The TSF shall drop and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;
b) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is a link-local address;
c) The TSF shall drop and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received.

FFW_RUL_EXT.1.8 The TSF shall process the applicable Stateful Traffic Filtering rules in an administratively defined order.

FFW_RUL_EXT.1.9 The TSF shall deny packet flow if a matching rule is not identified.

FFW_RUL_EXT.1.10 The TSF shall be capable of limiting an administratively defined number of *half-open TCP connections. In the event that the configured limit is reached, new connection attempts shall be dropped and the drop event shall be [logged]*.

5.3 TOE SFR Dependencies Rationale for SFRs

The collaborative Protection Profile for Stateful Traffic Filter Firewalls contains all the requirements claimed in this Security Target. As such, the dependencies are not applicable since the PP has been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the collaborative Protection Profile for Stateful Traffic Filter Firewalls which are derived from Common Criteria Version 3.1, Revision 4. The assurance requirements are summarized in the table below.

Assurance Class	Components	Components Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Life Cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM Coverage
Security Target evaluation	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_TSS.1	TOE summary specification
Tests	ATE_IND.1	Independent Testing – Conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

Table 10 Security Assurance Requirements

5.5 Rationale for Security Assurance Requirements

The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of

the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.

5.6 Assurance Measures

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by the vendor to satisfy the assurance requirements. The table below lists the details.

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes).
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked and how potential changes are incorporated.
ALC_CMS.1	
ATE_IND.1	SonicWALL, Inc. will provide the TOE for testing.
AVA_VAN.1	SonicWALL, Inc. will provide the TOE for testing.

Table 11 TOE Security Assurance Measures

6 TOE Summary Specification

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

Requirement	Rationale
FAU_GEN.1 FAU_GEN.2	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, VPN activity and other events related to the firewall. 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 the audit records are described in the guidance document. This includes administrator login and management activities associated with cryptographic keys. The logs do not contain the cryptographic keys. The SonicWall device can be configured to log network traffic associated with the rules set for allowing or denying particular packets. To do this, the administrator performs the following steps: • Under System > Administration, go to 'Enhanced Audit Logging Support' and enable the associated checkbox • Go to Log > Settings • Go to Network > Network Access and find 'Packet Allowed'. Select the checkbox next to 'Display Events in Log Monitor' • Select 'Apply' All packets that enter the SonicWall device are assessed according to the configured rules. A log is created any time a packet is dropped because it does not match an access rule. If the interface is overwhelmed, the packet will be dropped even if it matches an access rule. The normal log entries associated with access rules are not made when packets are dropped due to an overwhelmed interface; instead log records that indicate packets where dropped on a specified interface because the interface was overwhelmed are generated. In the case of key related operations, the name of the certificate the key is associated with is logged and used as the unique reference identifier.
FAU_STG_EXT.1	This SFR applies to the audit records for both FAU_GEN.1. 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 flash. Access to these records is restricted to authorized administrators with the appropriate privilege. Users who do not have the required privilege are not able to access the audit records.
FCS_CKM.1 FCS_CKM.2	The TOE supports Rivest-Shamir-Adleman (RSA) using 2048-bit keys, ECDSA using P-256 or P-384 keys, and Diffie-Hellman Group 14. RSA is used in support of TLS and IPsec. RSA and ECDSA keys are generated in accordance with FIPS PUB 186-4. The TOE complies with the requirements in FIPS PUB 186-4, Appendix B as described in Table 13. Diffie-Hellman Group 14 keys are generated using the parameters specified in RFC 3526 Section 3. The TOE performs Elliptic-Curve Diffie-Helman and Diffie-Hellman Group 14 to

	establish IPsec keys (FCS_IPSEC_EXT.1) for secure communications with VPN clients, VPN gateways, and the audit server. The TOE implements PKCS1_v1.5 conformant RSA-based key establishment scheme for asymmetric key establishment used in TLS (FCS_TLSS_EXT.1) for remote administration. The relevant CAVP certificate numbers are listed in Table 3.
FCS_CKM.4	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. Table 14 below shows the origin, storage location and destruction details for all plaintext keys. Unless otherwise stated, the keys are generated by the TOE. The SonicWall key used to verify firmware updates supports ECDSA (P-256 NIST curve). 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 zeroizes all keys, including those stored in the flash memory.
FCS_COP.1/DataEncryption	The TOE provides AES encryption/decryption in CBC and GCM modes with 128-bit and 256-bit keys.
FCS_COP.1/SigGen	The TOE supports signature generation and verification for RSA (2048 bits) and ECDSA (P-256, P-384), in accordance with FIPS PUB 186-4. RSA and ECDSA are used in IKE authentication. ECDSA is used to verify the signature on firmware updates.
FCS_COP.1/Hash	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 in support of TLS. SHA-256, SHA-384, and SHA-512 are used in support of IPsec. SHA-256 is used with ECDSA for the verification of firmware.
FCS_COP.1/KeyedHash	The TOE implements HMAC message authentication. HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 are supported with cryptographic key sizes of 160, 256, 384, and 512 bits and message digest sizes of 160, 256, 384, and 512 bits. HMAC-SHA-1 and HMAC-SHA-256 use a block size of 512-bits. HMAC-SHA-384 and HMAC-SHA-512 use a block size of 1024 bits.
FCS_HTTPS_EXT.1 FCS_TLSS_EXT.1	The TLS Server protocol is implemented in support of the HTTPS connection to the administrative interface. The TOE is always the received of connections. 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 (i.e. SSL 2.0, SSL 3.0, and TLS 1.0) requests will be denied. RSA with 2048-bit keys is implemented in these ciphersuites.

FCS_IPSEC_EXT.1	The TOE Administrator implements an IPsec policy to encrypt data between the TOE and the audit server. In general, an IPsec policy can 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 Protection Profile 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. The TOE supports IKE version 2. IKEv2 is the default proposal type for new VPN policies. Child SAs can be created, modified, and deleted independently at any time during the life of the VPN tunnel. IKEv2 initializes a VPN tunnel with a pair of message exchanges (two message/response pairs). • Initialize communication: The first pair of messages (IKE_SA_INIT) negotiate cryptographic algorithms, exchange nonces (random values generated and sent to guard against repeated messages) and perform a public key exchange. o Initiator sends a list of supported cryptographic algorithms, public keys, and a nonce. o Responder sends the selected cryptographic algorithm, the public key, a nonce, and an authentication request. • Authenticate: The second pair of messages (IKE_AUTH) authenticate the previous messages, exchange identities and certificates, and establish the first CHILD_SA. Parts of these messages are encrypted and integrity protected with keys established through the IKE_SA_INIT exchange, so the identities are hidden from eavesdroppers and all fields in all the messages are authenticated. o Initiator sends identity proof, such as a shared secret or a certificate, and a request to establish a child SA. o Responder sends the matching identity proof and completes negotiation of a child SA. This exchange consists of a single request/response pair. It may be initiated by either end of the SA after the initial exchanges are completed. All messages following the initial exchange are cryptographically protected using the cryptographic algorithms and keys negotiated in the first two messages of the IKE exchange. Either endpoint can initiate a CREATE_CHILD_SA exchange, so in this section the term “initiator” refers to the endpoint initiating this exchange. o The Initiator sends a child SA offer and, if the data is to be encrypted, the encryption method and the public key. o The Responder sends the accepted child SA offer and, a public key.

	The TOE administrative interface provides a VPN Policies page on which the policies applicable to a particular VPN can 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. Site to Site Policies apply when the device acts as a remote client headend. In this case, the IPsec Primary Gateway Name or Address is set to 0.0.0.0. On the Network tab, the Administrator selects 'Use IKEv2 IP pool'. The pool is created with the addresses that are to be provided to the remote clients. Any required third-party certificates would have to be loaded on the VPN clients. The TOE can be only operated in Tunnel mode in the evaluated configuration. This is a default setting and cannot be changed when using IKEv2. AES-CBC-128, AES-CBC-256, AES-GCM-128, and AES-GCM-256 are supported for ESP. The HMAC implementation conforms to HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512. The IKE payload is encrypted using AES-CBC-128 or AES-CBC-256. The IKEv2 SA lifetime is selected in the SPD and can be set to be between 120 and 86400 seconds (24 hours). The IKEv2 Child SA lifetime is selected in the SPD and can also be set to be between 120 and 28800 seconds (8 hours). The TOE supports Group 14, 256-bit Random ECP Group (Group 19) and 384-bit Random ECP Group (Group 20). The DRBG described in FCS_RBG_EXT.1 is used to generate each nonce for DH groups 14, 19, and 20 for IKEv2. The TOE supports SHA-256, SHA-384, and SHA-512 as the hash in the PRF. The size of the nonce is 128-256 bits (half of the pseudorandom function with a minimum of 128 bits). The symmetric algorithms supported for IKEv2 IKE_SA uses the same or greater key length as the symmetric algorithms used to protect IKEv2 CHILD_SA. The available options ensure that the IKEv2 IKE_SA symmetric algorithm key length is equal to or greater than the IKEv2 CHILD_SA symmetric algorithm key length. Peer authentication is performed using third-party RSA or ECDSA certificates. Reference identifiers are supported for SAN: IP address, SAN: Fully Qualified Domain Name (FQDN), SAN: user FQDN, and Distinguished Name (DN). 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.
FCS_RBG_EXT.1	The TOE implements a DRBG in accordance with ISO/IEC 18031:2011 using Hash_DRBG. The DRGB is seeded using 880-bits from a third-party entropy source provided by the Cavium Octeon hardware on the hardware appliances. The third-party entropy source is assumed to have at least .5 bits of entropy per byte, so the DRBG is seeded with at least 256 bits of entropy. The entropy source is discussed in more detail in the Entropy documentation.
FDP_RIP.2	The TOE ensures that no data is reused with processing network packets. Once packets have been sent from the TOE, the memory buffers are allocated to the buffer pool. When memory is returned to the buffer pool, the memory is overwritten with pseudo random data. The cleared memory can then be reallocated in support of the next request.

FIA_AFL.1	The SonicWall appliance can be configured to lockout an administrator on the remote administration interface if incorrect login credentials are provided. This is configured using the Enable Administrator/User Lockout features. The number of failed attempts per minute before lockout can be set. The Lockout period, which is the time that must elapse before the user is allowed to attempt to login again, can also be set. If a user enters the configured number of incorrect login credentials, the user is blocked from submitting additional credentials until the lockout period has expired.
FIA_PMG_EXT.1 FIA_UIA_EXT.1 FIA_UAU_EXT.2 FIA_UAU.7	The SonicOS Management UI is the application used to manage the TOE devices. It is protected by HTTPS. A directly connected serial console provides a local text-based interface to manage the TOE. A management 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 for the SonicOS Management UI and local console both require 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. The TOE hashes the user entered password and compares it to the stored hash for the associated username. The authentication is considered successful and access is granted if the hashes match. If unsuccessful, the logon screen will be displayed. No functionality is available prior to login other than viewing the previously mentioned warning banner. Passwords must meet the rules set by the administrator. These rules are governed by the requirements described in FIA_PMG_EXT.1. Minimum password lengths are configurable for 1 to 99 characters.
FIA_X509_EXT.1/Rev FIA_X509_EXT.3	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 validity dates • the validation path, ensuring that the certificate path terminates with a trusted CA certificate • basicConstraints, ensuring the presence of the basicConstraints extension • revocation status, using OCSP • extendedKeyUsage properties, if the certificate is used for 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. When the certificate is used, the OCSP server is contacted to verify that the certificate is still valid. If the validity of a certificate cannot be verified, the system rejects the certificate and drops the connection.
FIA_X509_EXT.2	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. Certificates are supplied back to the clients (the TOE only acts as the receiver of connections) and client certificates are neither required nor validated.

	If the validity of a certificate cannot be verified, the system rejects the certificate and drops the connection.
FMT_MOF.1/Services FMT_MOF.1/ManualUpdate FMT_MTD.1/CryptoKeys FMT_MTD.1/CoreData FMT_SMF.1 FMT_SMR.2	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. Rules for VPN traffic are configured through the Firewall Access Rules. The Administrator navigates to Firewall > Access Rules and selects the 'Matrix' checkbox. Under 'Zones', the Administrator selects VPN to LAN, WAN or VPN and then configures the rules. This will configure rules specifically for the VPN traffic.
FPT_APW_EXT.1	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.
FPT_SKP_EXT.1	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.
FPT_STM_EXT.1	The TOE provides reliable time stamps that are used for audit records. 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. 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.

FPT_TST_EXT.1
FPT_TST_EXT.3
	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
	• 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 • ECDSA Known Answer Test • ECDSA Signature and Verification Known Answer Tests When a new firmware image is loaded, the cryptographic module verifies the ECDSA signed SHA-256 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.
FPT_TUD_EXT.1	TSF software can 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 loaded and an error appears. Firmware can be uploaded, but not activated. The new firmware will not be activated until the administrator boots the device with the new firmware by selecting the new firmware and 'Boot'. The version of firmware running may be queried through the TOE UI. The version of the most recently installed firmware may also be queried through the TOE UI.
FTA_SSL_EXT.1 FTA_SSL.3 FTA_SSL.4 FTA_TAB.1	All access to the TOE takes place through the web-based management interface over HTTPS or local serial console. The web-based management interface can be accessed using the GUI (Note that the Getting Started or Quick Start Guide refers to the GUI as the MGMT interface). 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, both local and remote, are presented with a Security Administrator-configured advisory notice and consent warning prior to TOE login.
FTP_ITC.1 FTP_TRP.1/Admin	IPsec VPN tunnels are used to provide a trusted communication channel between the TOE and the external audit server and to support VPN communications. 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. The TOE supports IKE version 2 in protecting these communications from disclosure and detecting modification.

	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 is used in the supported TLS ciphersuites.
FFW_RUL_EXT.1	Packets are received by the SonicWall device on one of three Ethernet links: the LAN, WAN, or optional DMZ link. The packets are analyzed in the communications stack at a level that is best described as above the Ethernet driver, but below the networking stack. Transport-and application-layer data is also examined. This higher-level data is used to provide the stateful inspection security. During this analysis, packets are modified, dropped, passed up to the networking stack, or rewritten directly to another Ethernet link, as appropriate. The analysis is based on a set of rules entered by the firewall administrator. The SonicWall device acts as a single component. If the component fails, processing ceases and all traffic is stopped. SonicWall interacts with the Ethernet drivers, and also with the networking stack. An incoming packet will initially be read by the Ethernet driver. At this point, the device does one of three things: • Drop the packet. It will do this based on the security policy configured by the administrator • Rewrite the packet, which may be modified, to another Ethernet link • Pass the packet up to the stack Conceptually, the stack exists on the LAN link of the SonicWall. If the stack tries to communicate with the DMZ or Internet, then the device will provide network address translation. When an Ethernet packet is received on a given link, Address Resolution Protocol (ARP) and Point-to-Point Protocol over Ethernet (PPPoE) packets are first vectored off to their respective handlers. IP packets are sent through a complicated series of code modules that analyze them, modify them, forward them, or drop them, as appropriate. The path of a packet through these code modules is described here. First, raw fields of the packet buffer are analyzed and unpacked into a machine-aligned structure. This is done for optimization; endian conversion and alignment shifting only happens once. Next, the packet goes through a sequence of stateless analysis. That is, the packet is analyzed based solely on the contents of the packet, not taking the connection into account. • IPSec packets are vectored to the IPSec handling code. This essentially encapsulates and encrypts (or unencapsulates and decrypts) the packet. Conceptually, the IPSec tunnel terminates on the inside of the firewall, so packets are encrypted before passing through the firewalling, content filtering, and other code. Conversely, incoming traffic is decrypted and then written to the LAN without filtration. • Stateless Attack Prevention analysis is performed. This consists of stateless checks for malformed and fragmented packets, smurf amplifiers, Layer 4 Denial of Service (LAND) attacks, etc. The analysis code may decide to drop the packet and create a log message. • Packets addressed to the firewall itself may be vectored off at this point. For instance, TCP packets directed to the management interface may be passed up the stack. Packets may be sent directly to code modules without depending on the stack. For example, UDP packets may be directed to the DHCP server or client.

	• DNS packets may be intercepted in order to support domain-name access to the firewall without configuration of a DNS server, and also to foil a bug with IE4 involving reverse-DNS lookups for java applets.
	• Packets may be bounced off the LAN interface if they have been routed improperly; ICMP redirect packets are sent in an attempt to rectify the problem.
	
	Next, the packet goes through a sequence of stateful analysis.
	
	• A connection cache lookup takes place. If a cache entry isn't found, one is added (even if this packet will be dropped).
	• Incoming packets must be NAT-remapped during this cache lookup process in order to find them properly. From this point on, the destination IP and port information will be remapped to internal, private values.
	• Stateful attack prevention is performed.
	  ○ SYN floods are detected, and any suspicious connections are reset. Technically, this step happens BEFORE the connection cache lookup. This is because SYN flood prevention uses a different cache than the main connection cache. This is mostly for historical reasons; it may be changed in the future. (In versions 1.x, there was no firewalling of the DMZ; only attack prevention).
	  ○ IP Spoof checking is simply a sanity check of the source and destination IP addresses against the static routing information in the box. This could be done statelessly, however, there is a significant speed advantage when cached routing information is used.
	  ○ TCP sequence numbers are offset by a random value for every distinct TCP connection.
	• Antivirus policing may redirect a web query to the Virus Update website if the client's antivirus software is out of date.
	• User-based authentication tables are checked; these may override packet filtration or content filtration.
	• Packet filtering rules are checked. If the packet matches an 'ALLOW' access rule, the connection cache is created. If the packet matches a 'DENY' rule, or there is no matched 'ALLOW' rule, the packet does not proceed.
	• Stateful inspection takes place. This is a set of application-specific code modules that examine application-layer packet contents in order to add 'anticipated' cache elements on the fly. In other words, a cache element will be added for a connection that would normally violate the packet filtering rules, such as an incoming FTP data connection. Since the cache element already exists by the time the first incoming SYN packet arrives, it will not be rejected by the packet filtration.
	• Content filtration takes place. This is primarily for Web traffic, although some filtration can be done on other protocols. Note that it is not sufficient to identify traffic using TCP port 80, since some web sites use non-standard ports. The SonicWall device checks for a 'GET /' command in the application-layer data.
	  ○ Cybernot list
	  ○ Trusted and forbidden domains
	  ○ ActiveX, Java, and Cookie blocking
	  ○ Keyword scanning
	  ○ Proxy servers blocking
	• License enforcement takes place. For instance, connections from the eleventh IP address on the LAN of a 10-user SOHO box will be rejected.
	• Outgoing packets are NAT-remapped. From this point on, the source IP and port information will be set to external, valid Internet values. (That is, unless the WAN port is on its own private network).

	• Proxy redirection may take place, if the firewall is configured to send all web traffic through an external proxy such as a web cache. This is done by prepending some data to pieces of the web command, and then changing the destination IP address to match the proxy server rather than the actual web server. Finally, the packet is written back to the network. The Ethernet link used to write the packet (LAN, WAN, or DMZ) is determined by the static routing information stored in the firewall's configuration. After the packet is written out, some cleanup takes place, and then the packet is done. If any component fails, packets will not be accepted into the connection cache, and will therefore not be allowed to flow through the device. The following RFCs are supported: • RFC 792 (ICMPv4) • RFC 4443 (ICMPv6) • RFC 791 (IPv4) • RFC 2460 (IPv6) • RFC 793 (TCP) • RFC 768 (UDP) Conformance to these RFCs is demonstrated by protocol compliance testing by the product QA team. The Stateful packet filtering policy consists of the following rules and attributes. • Action: (Allow/Deny/Discard) o Configure to permit or drop the packet • From: (Zone/Interface) o Packet ingress point • To: (Zone/Interface) o Packet egress point • Source Port: (Services Object) o The protocol and the source port of the packet • Services: (Services Object) o The protocol and the destination port of the packet • Source: (Host/Range/Network) • Source IP: The source IP of the packet • Destination: (Host/Range/Network) • Destination IP: the Destination IP of the packet • Enable Logging (Checkbox) • Log the action when it is taking place • TCP Connection Inactivity Timeout (minutes) • UDP Connection Inactivity Timeout (seconds) The attributes are all configurable for ICMPv4, ICMPv6, IPv4, IPv6, TCP and UDP policies. Logging can be configured for each access rule. The source and destination address are configurable for each access rule. The supported header fields for IPv4, IPv6, TCP, UDP, ICMPv4 and ICMPv6 are listed below in Table 15. Stateful session handling is supported for TCP and UDP. Source and destination addresses, and source and destination ports are used together to recognize TCP flow in support of stateful session handling. Sequence numbers are used to

	ensure that the received data falls within the window defined for the protocol. Flags are used to track the connection against the defined TCP State Machine states:
	• Listen State: Only a TCP packet with just the SYN flag is considered valid.
	• Syn-Sent State:
	  ○ ACK number (if present) must be valid.
	  ○ RST packet (with a valid TCP ACK number) is valid.
	  ○ FIN packet (which does not have the SYN bit set) is also considered valid.
	• Syn-Received, Established, Fin-Sent, and Fin-Acked States:
	  ○ SEQ number must be within the TCP window for the destination or be that for Keep-Alive packet.
	  ○ RST packet (with a valid TCP SEQ number) is valid.
	  ○ ACK number must also be present and valid in this state.
	  ○ A SYN seen in this state will cause the TCP connection to be closed.
	• Close-Wait State:
	  ○ A SYN is valid (to re-open the same TCP connection).
	  ○ Any other packet which is also valid in the previous state is acceptable.
	For UDP, source and destination addresses, and source and destination ports are used together to be checked to match with an access rule. Following a UDP request, the TOE will accept return packets for a configurable period of time. This is generally in the order of several seconds and is configurable as the UDP Timeout in the applicable access rule.
	Stateful sessions are removed when complete, or when the timeout is triggered.
	For TCP connection completion, the connection is closed in one of two ways:
	• Syn-Sent State
	  ○ A validated RST will cause the action of the TCP connection to be closed.
	• Syn-Received, Established, Fin-Sent, Fin-Acked, and Close-Wait States
	  ○ A validated RST will cause the action of the TCP connection to be closed.
	  ○ Acknowledged TCP FINs will cause the action of the TCP connection to be closed.
	Session removal becomes effective immediately after Connection cache is removed.
	Each packet flow through the TOE triggers a timestamp update to its connection cache. The TOE checks this timestamp, and if the connection cache timeout has been reached, the session is removed.
	The TOE will automatically drop and log the event when the following is found:
	• A packet is found to be an invalid fragment. A fragment is determined to be invalid if it cannot be combined with other fragments to form a packet. The offset may be incorrect, or it may be considered to be too small
	• A fragment cannot be completely re-assembled
	• A packet with a source address that is defined as being on a broadcast network
	• A packet with a source address that is defined as being on a multicast network
	• A packet with a source address that is defined as being a loopback address
	• A packet with a source or destination address that is defined as unspecified or reserved for future use
	• A packet with a source or destination address that is defined as an unspecified address or an address reserved for future definition and use
	• A packet with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified

	The algorithm applied to incoming packets performs the following actions:
	• In the evaluated configuration, the default action is to DENY a packet. The TOE checks the incoming packet against all of the access rules. If the packet does not match any access rule and does not belong to an approved established connection, then the default action is to DENY the packet.
	• The TOE performs a Connection cache lookup
	  o each connection cache represents an established session
	  o For incoming packets, srcIp, dstIp, srcPort, dstPort, ipType are used together as a hash index to find the matched connection cache
	  o An access rule check is performed if the connection cache lookup fails
	• The TOE performs an access rule check only if the connection cache lookup fails. The following rules are applied in an access rule check:
	  o Access rules are ordered by Priority. The rule with higher Priority will be applied
	  o For incoming packets, srcZone, dstZone, srcIp, dstIp, srcPort, dstPort, ipType are used together as a hash index to find the matching access rule
	  o If an incoming packet matches an access rule with the ALLOW action, a new connection cache is added. Otherwise the packet is dropped
	In the evaluated configuration, the default action is to DENY a packet if the packet does not match any of the access rules. However, this does not apply for dynamic protocol traffic.
	Dynamic protocols include ftp, tftp, pptp, and oracle. These protocols are similar to ftp in that they use multiple TCP connections. The first connection is the control connection. A particular command, specified by the protocol, opens the one or more additional data connections. The TOE inspects the control connection to find the target commands and adds the new connection cache appropriate to allow the network traffic.
	The TOE tracks and maintains information relating to the number of half-open TCP connections as follows:
	• There is an administratively defined limit for half-open TCP connections based on:
	  o TCP Handshake Timeout (seconds)
	  o Maximum Half Open TCP Connections
	• There is a TCP Handshake Timeout (seconds)
	  o Each half-open TCP connection is removed if the handshake is not complete by the time this timeout is reached
	• There is a maximum number of allowable Half Open TCP Connections
	  o A global counter is used by the TOE to track the number of all half-open TCP connections. When this number reaches the value of Maximum Half Open TCP Connections, new incoming TCP connections are dropped

Table 12 TOE Summary Specification SFR Description

FIPS 186-4 Appendix B-3 Section	Compliance
B.3.1	All shall statements met. In accordance with the reference, p and q with length of 512 are not generated using the described methods.
B.3.6	All shall statements met.
B.4	All shall statements met.

Table 13 FIPS 186-4 Compliance

Type/ Description	Generation/ Algorithm	Storage	Destruction Method
RSA private key used for TLS	RSA (2048 bits)	Stored in flash memory Held in the RAM buffer in plaintext	The 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 in flash memory Held in the RAM buffer in plaintext	The 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
Key Agreement Keys used for IPsec	DH (2048 bits) ECDH (P-256, P-384)	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 The plaintext key is overwritten with a pseudo-random pattern upon termination of the session or reboot of the appliance
Authentication Keys used for IPsec	RSA (2048 bits) ECDSA (P-256, P-384)	Stored in flash memory Held in the RAM buffer in plaintext	The 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 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)	Stored in Flash Memory	The key may be overwritten by a software update

Table 14 Key Material

Protocol	Field	Configuration Support
IPv4	Header Length	1. Single Numeric Value 2. Range of Numeric Values

Protocol	Field	Configuration Support
	Packet Length	1. Single Numeric Value 2. Range of Numeric Values
	Identity	1. Single Numeric Value 2. Range of Numeric Values
	IP Flags	Selection: 1. dont-fragment (0x4) 2. more-fragments (0x2) 3. reserved (0x8).
	Fragment Offset	1. Single Numeric Value 2. Range of Numeric Values
	TTL	1. Single Numeric Value 2. Range of Numeric Values
	Protocol	Single Numeric Value
	Header Checksum	Selection: 1. valid 2. Invalid
	IP Options	Selection: 1. (7) record-route 2. (68) timestamp 3. (130) security 4. (131) loose-source-route 5. (136) stream-id 6. (137) strict-source-route 7. (148) router-alert
IPv6	Traffic Class	Selection: 1. (10) af11 2. (12) af12 3. (14) af13 4. (18) af21 5. (20) af22 6. (22) af23 7. (26) af31 8. (28) af32 9. (30) af33 10. (34) af41 11. (36) af42 12. (38) af43 13. (46) ef
	Flow Label	1. Single Numeric Value 2. Range of Numeric Values
	Payload Length	1. Single Numeric Value 2. Range of Numeric Values
	Next Header	Selection: 1. (0) hop-by-hop 2. (1) icmp

Protocol	Field	Configuration Support
		3. (2) igmp 4. (4) ipip 5. (6) tcp 6. (8) egp 7. (17) udp 8. (41) ipv6 9. (43) routing 10. (44) fragment 11. (46) rsvp 12. (47) gre 13. (50) esp 14. (51) ah 15. (58) icmpv6 16. (59) no-next-header 17. (60) dstops 18. (89) ospf 19. (103) pim 20. (112) vrrp 21. (132) sctp 22. (135) mobility 23. (201) home address
	Hop Limit	1. Single Numeric Value 2. Range of Numeric Values
TCP	Header Length	1. Single Numeric Value 2. Range of Numeric Values
	Packet Length	1. Single Numeric Value 2. Range of Numeric Values
	Flags	Selection: 1. (0x01) fin 2. (0x02) syn 3. (0x04) rst 4. (0x08) push 5. (0x10) ack 6. (0x20) urgent
	Option	Selection: 1. (0)End of option list 2. (1)No-Operation 3. (2)Maximum Segment Size 4. (3)Window Scale 5. (8)Timestamps
	Checksum	Selection: 1. valid 2. invalid
	Urgent Pointer	1. Single Numeric Value 2. Range of Numeric Values

Protocol	Field	Configuration Support
UDP	Length	1. Single Numeric Value 2. Range of Numeric Values
	Checksum	Selection: 1. valid 2. Invalid
ICMPv4	Type	Selection: 1. (0)echo-reply 2. (3)unreachable 3. (4)source-quench 4. (5)redirect 5. (8)echo-request 6. (9)router-advertisement 7. (10)router-solicit 8. (11)time-exceeded 9. (12)parameter-problem 10. (13)timestamp 11. (14)timestamp-reply 12. (15)info-request 13. (16)info-reply 14. (17)mask-request 15. (18)mask-reply
	Code	Selection: 1: parameter-problem: (1) required-option-missing 2: parameter-problem: (0) ip-header-bad 3: redirect: (1) redirect-for-host 4: redirect: (0) redirect-for-network 5: redirect: (1) redirect-for-host 6: redirect: (2) redirect-for-tos-and-net 7: redirect: (3) redirect-for-tos-and-host 8: time-exceeded: (0) ttl-eq-zero-during-transit 9: time-exceeded: (1) ttl-eq-zero-during-reassembly 10: unreachable: (0) network-unreachable 11: unreachable: (1) host-unreachable 12: unreachable: (3) port-unreachable 13: unreachable: (4) fragmentation-needed 14: unreachable: (6) destination-network-unknown 15: unreachable: (7) destination-host-unknown 16: unreachable: (9) destination-network-prohibited 17: unreachable: (10) destination-host-prohibited 18: unreachable: (11) network-unreachable-for-TOS 19: unreachable: (12) host-unreachable-for-TOS 20: unreachable: (13) communication-prohibited-by-filtering 21: unreachable: (14) host-precedence-violation 22: unreachable: (15) precedence-cutoff-in-effect
	Header Checksum	Selection: 1. valid 2. Invalid

Protocol	Field	Configuration Support
ICMPv6	Type	Selection: 1. (1) destination-unreachable 2. (2) packet-too-big 3. (3) time-exceeded 4. (4) parameter-problem 5. (100) private-experimentation-100 6. (101) private-experimentation-101 7. (128) echo-request 8. (129) echo-reply 9. (130) membership-query 10. (131) membership-report 11. (132) membership-termination 12. (133) router-solicit 13. (134) router-advertisement 14. (135) neighbor-solicit 15. (136) neighbor-advertisement 16. (137) redirect 17. (138) router-renumbering 18. (139) node-information-request 19. (140) node-information-reply 20. (141) inverse-neighbor-discovery-solicitation 21. (142) inverse-neighbor-discovery-advertisement 22. (144) home-agent-address-discovery-request 23. (145) home-agent-address-discovery-reply 24. (146) mobile-prefix-solicitation 25. (147) mobile-prefix-advertisement-reply 26. (148) certificate-path-solicitation 27. (149) certificate-path-advertisement 28. (200) private-experimentation-200 29. (201) private-experimentation-201
	Code	Selection: 1. parameter-problem: (0) ip6-header-bad 2. parameter-problem: (1) unrecognized-next-header 3. parameter-problem: (2) unrecognized-option 4. time-exceeded: (0) ttl-eq-zero-during-transit 5. time-exceeded: (1) ttl-eq-zero-during-reassembly 6. destination-unreachable: (0) no-route-to-destination 7. destination-unreachable: (1) administratively-prohibited 8. destination-unreachable: (3) address-unreachable 9. destination-unreachable: (4) port-unreachable
	Header Checksum	Selection: 1. valid 2. Invalid

Table 15 Supported Header Fields for Firewall Filtering

Protocol	Header Field	Data Elements
ICMPv4	Type	(0) echo-reply (3) unreachable (4) source-quench

Protocol	Header Field	Data Elements
		(5) redirect (8) echo-request (9) router-advertisement (10) router-solicit (11) time-exceeded (12) parameter-problem (13) timestamp (14) timestamp-reply (15) info-request (16) info-reply (17) mask-request (18) mask-reply
	Code	Parameter-problem: (1) required-option-missing parameter-problem: (0) ip-header-bad redirect: (1) redirect-for-host redirect: (0) redirect-for-network redirect: (2) redirect-for-tos-and-net redirect: (3) redirect-for-tos-and-host time-exceeded: (0) ttl-eq-zero-during-transit time-exceeded: (1) ttl-eq-zero-during-reassembly unreachable: (0) network-unreachable unreachable: (1) host-unreachable unreachable: (3) port-unreachable unreachable: (4) fragmentation-needed unreachable: (6) destination-network-unknown unreachable: (7) destination-host-unknown unreachable: (9) destination-network-prohibited unreachable: (10) destination-host-prohibited unreachable: (11) network-unreachable-for-TOS unreachable: (12) host-unreachable-for-TOS unreachable: (13) communication-prohibited-by filtering unreachable: (14) host-precedence-violation unreachable: (15) precedence-cutoff-in-effect
	Header Checksum	Hex value
	Rest of Header	Contains the data specific to the message type indicated by the Type and Code fields
ICMPv6	Type	(1) destination-unreachable (2) packet-too-big (3) time-exceeded (4) parameter-problem (100) private-experimentation-100 (101) private-experimentation-101 (128) echo-request (129) echo-reply (130) membership-query (131) membership-report (132) membership-termination (133) router-solicit (134) router-advertisement (135) neighbor-solicit (136) neighbor-advertisement (137) redirect (138) router-renumbering

Protocol	Header Field	Data Elements
		(139) node-information-request (140) node-information-reply (141) inverse-neighbor-discovery-solicitation (142) inverse-neighbor-discovery-advertisement (144) home-agent-address-discovery-request (145) home-agent-address-discovery-reply (146) mobile-prefix-solicitation (147) mobile-prefix-advertisement-reply (148) certificate-path-solicitation (149) certificate-path-advertisement (200) private-experimentation-200 (201) private-experimentation-201
	Code	parameter-problem: (0) ip6-header-bad parameter-problem: (1) unrecognized-next-header parameter-problem: (2) unrecognized-option time-exceeded: (0) ttl-eq-zero-during-transit time-exceeded: (1) ttl-eq-zero-during-reassembly destination-unreachable: (0) no-route-to-destination destination-unreachable: (1) administrativelyprohibited destination-unreachable: (3) address-unreachable destination-unreachable: (4) port-unreachable
	Header Checksum	Hex value
TCP	Header Length	Single Numeric Value Range of Numeric Values
	Packet Length	Single Numeric Value Range of Numeric Values
	Flags	(0x01) fin (0x02) syn (0x04) rst (0x08) push (0x10) ack (0x20) urgent
	Option	(0) End of option list (1) No-operation (2) Maximum Segment Size (3) Window Scale (8) Timestamps
	Checksum	Hex value
	Urgent Pointer	Single Numeric Value Range of Numeric Values
	Source Port	Source port number
	Destination Port	Destination port number

Protocol	Header Field	Data Elements
	Acknowledgement Number	Next sequence number value
	Reserved	Must be zero
	Offset	The number of 32 bit words in the TCP Header
	Window	Number of octets
UDP	Length	Single numeric value Range of numeric values
	Checksum	Hex value
	Source Port	Source port number
	Destination Port	Destination port number
IPv4	Version	Four bit field equal to 4
	Header Length	Single Numeric Value Range of Numeric Values
	Packet Length	Single Numeric Value Range of Numeric Values
	Identity	Single Numeric Value Range of Numeric Values
	IP Flags	Don't-fragment (0x4) More-fragments (0x2) Reserved (0x8)
	Fragment Offset	Single Numeric Value Range of Numeric Values
	TTL	Single Numeric Value Range of Numeric Values
	Protocol	Single Numeric Value
	Header Checksum	Hex value
	Source Address	IP address of sending node
	Destination Address	IP address of intended receiving node
	IP Options	(7) record-route (68) timestamp (130) security (131) loose-source-route (136) stream-id

Protocol	Header Field	Data Elements
		(137) strict-source-route (148) router-alert
IPv6	Version	Four bit field equal to 6
	Traffic Class	(10) af11 (12) af12 (14) af13 (18) af21 (20) af22 (22) af23 (26) af31 (28) af32 (30) af33 (34) af41 (36) af42 (38) af43 (46) ef
	Flow Label	Single Numeric Value Range of Numeric Values
	Payload Length	Single Numeric Value Range of Numeric Values
	Next Header	(0) hop-by-hop (1) icmp (2) igmp (4) ipip (6) tcp (8) egp (17) udp (41) ipv6 (43) routing (44) fragment (45) rsvp (47) gre (50) esp (51) ah (58) icmpv6 (59) no-next-header (60) dstops (89) ospf (103) pim

Protocol	Header Field	Data Elements
		(112) vrrp (132) sctp (135) mobility (201) home address
	Hop Limit	Single Numeric Value Range of Numeric Values
	Source Address	IP address of sending node
	Destination Address	IP address of intended receiving node
	Routing Header	(0) Source Route (1) Nimrod (2) Type 2 Routing Header (3) RPL Source Route Header

Table 16 Packet Header Payload Inspection Elements