Stonesoft Next Generation Firewall (NDPP11e3/STFFEP10) Security Target

Version 0.7
February 29, 2016

Prepared for:

Forcepoint LLC
10900-A Stonelake Blvd.
Austin, TX 78759, USA

www.Forcepoint.com

Prepared By:

www.gossamersec.com

Table of Contents

LIST OF TABLES

1 Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is the Stonesoft Next Generation Firewall provided by FORCEPOINT. The TOE is being evaluated as a network infrastructure device.

The Security Target contains the following additional sections:

- Conformance Claims (Section 2)
- Security Objectives (Section 3)
- Extended Components Definition (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a letter placed at the end of the component. For example FDP_ACC.1a and FDP_ACC.1b indicate that the ST includes two iterations of the FDP_ACC.1 requirement, a and b.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- The NDPP uses an additional convention – the 'case' – which defines parts of an SFR that apply only when corresponding selections are made or some other identified conditions exist. Only the applicable cases are identified in this ST and they are identified using **bold** text.
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

Terminology

This following acronyms and terms are used throughout this document.

AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CC	Common Criteria for Information Technology Security Evaluation
CLI	Command Line Interface
ECDSA	Elliptic Curve Digital Signature Algorithm

FDP	User Data Protection CC Class
FIA	Identification and Authentication CC Class
FIPS	Federal Information Processing Standard
FMT	Security Management CC Class
FTP	File Transfer Protocol
HTTP	Hyper Text Transfer Protocol
ICMP	Internet Control Message Protocol
MOF	Management of Functions
MTD	Management of TSF Data
NDPP11e3	Protection Profile for Network Devices, Version 1.1 with Errata #3
NGFW	Next Generation Firewall
NIAP	National Information Assurance Partnership
PP	Protection Profile
RFC	Request for Comments
RSA	Rivest, Shamir and Adleman
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SMC	Security Management Center
ST	Security Target
STFFEP10	Extended Package Stateful Traffic Filter Firewall, Version 1.0
TLS	Transport Layer Security
TOE	Target of Evaluation
TSC	TOE Scope of Control
TSF	TOE Security Function
TSP	TOE Security Policy
UAU	User Authentication
UDP	User Datagram Protocol

1.1 Security Target Reference

ST Title – Stonesoft Next Generation Firewall (NDPP11e3/STFFEP10) Security Target

ST Version – Version 0.7

ST Date – February 29, 2016

1.2 TOE Reference

TOE Identification – Forcepoint^{TM} Stonesoft Next Generation Firewall composed of the NGFW Engine (version 5.10.1) and Security Management Center (SMC) Appliance (version 5.10.0 with SMC Appliance patch 5.10.0P001). The NGFW Engine is evaluated on the following models using only Ethernet networking[1]:

Firewall Appliances:

Rack Mounted Firewall models

- 1035
- 1065
- 1401
- 1402
- 3202 (2U)
- 3207 (2U)
- 3206 (2U)

[1] The wireless networking supported by some models of Firewall Appliance was not included in the evaluation.

- 3301 (2U)
- 5206 (3U)

Desktop Firewall models

- 320X-C1
- 321-C2
- 325-C2

TOE Developer – Forcepoint LLC

Evaluation Sponsor – Forcepoint LLC

1.3 TOE Overview

The Target of Evaluation (TOE) is Stonesoft Next Generation Firewall (NGFW) version 5.10.1 and Security Management Center (SMC) Appliance version 5.10.0 with SMC Appliance patch 5.10.0P001.

The Stonesoft Next Generation Firewall is a stateful packet filtering firewall. Being a stateful packet filtering firewall, the NGFW filters network traffic optimized through the use of stateful packet inspection. The NGFW is intended to be used as a network perimeter security gateway that provides a controlled connection. The NGFW is centrally managed and generates audit records for security critical events.

1.4 TOE Description

The Stonesoft Next Generation Firewall is a stateful packet filtering firewall. The Stonesoft Next Generation Firewall (NGFW) system is composed of two physical appliances: the NGFW engine and the Security Management Center (SMC) Appliance. The NGFW engine controls connectivity and information flow between internal and external connected networks. The SMC Appliance provides administrative functionality supporting the configuration and operation of one or more NGFW engines. Throughout the remainder of this document, references to the NGFW engine are meant to reference the firewall engine as a TOE component, while references to the NGFW are meant to refer to the TOE as a whole.

The NGFW engine controls connectivity and information flow between internal and external connected networks. The NGFW engine also provides a means to keep the internal host’s IP-address private from external users. The NGFW engine is intended to be used as a network perimeter security gateway that provides a controlled connection.

The NGFW is assumed to be installed and operated within a physically protected environment, administered by trusted and trained administrators over a trusted and separate management network. Multiple installations of the NGFW engine may be used in combination to provide a company with an overall network topology.

The NGFW engine runs on a hardened Linux operating system[2] that is shipped with the product. The software (which is also part of the NGFW engine product) runs on a single or multi-processor Forcepoint platform.

The SMC appliance – a management system comprising a Management Server, Log Server and McAfee Linux Operating System (MLOS) to support the management and operation of the firewall – is included as part of the product. The MLOS that is used for the management server is the same underlying OS that is used in several other evaluated security products and has undergone prior evaluation as part of those products.

1.4.1 TOE Architecture

The Stonesoft Next Generation Firewall (NGFW) system is composed of two physical appliances: the NGFW engine and the Security Management Center (SMC) Appliance. The NGFW engine is an appliance composed of firewall functionality, Engine OpenSSL Library and a Linux operating system. The SMC Appliance is composed of two custom built Java applications called the Management Server and the Log Server, running on the McAfee Linux Operating System (MLOS, version 2.2.3) with support from OpenSSL and a Java runtime environment.

[2] Based on Linux kernel 3.16.

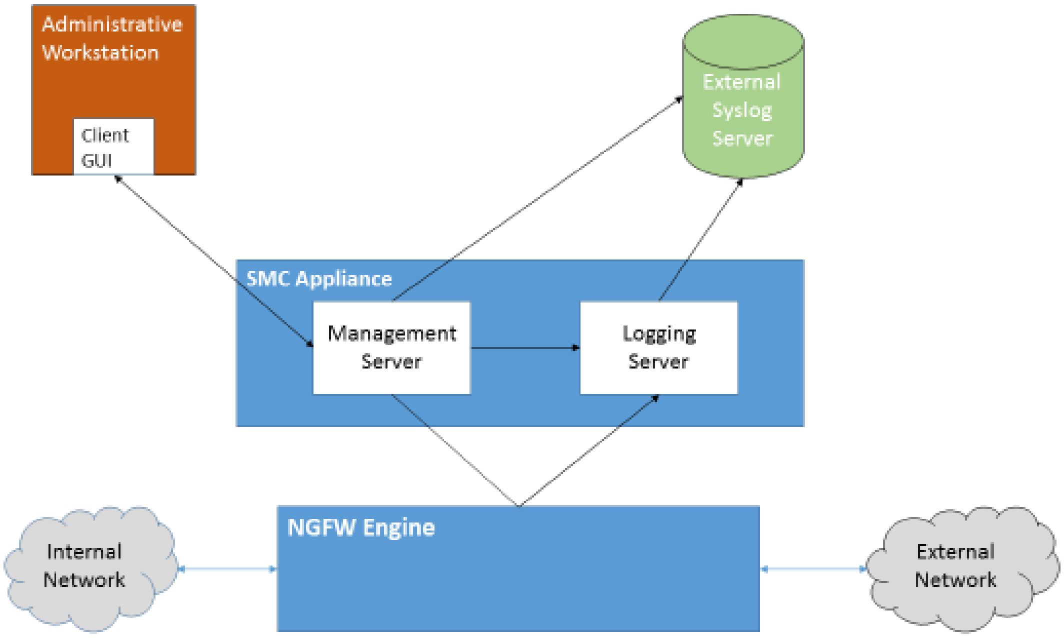

Figure 11 TOE Components, Communication Paths and IT Environment.

The NGFW engine (a.k.a., the engine) is responsible for performing all firewall packet handling, analysis and filtering that is provided by the NGFW system.

The Management server on the SMC appliance provides the majority of the administrative capabilities in the NGFW system. A very limited console interface is provided on the SMC appliance by the MLOS and used to verify and update TOE software.

The NGFW engine does not have a local administrative interface, and can be configured only by an SMC appliance. The Management server is responsible for transferring the administrator defined configuration to the NGFW engine as the administrator makes configuration changes (these configuration changes are known as a 'security policy').

The log server in the SMC appliance is responsible for collecting audit events from other components (NGFW engines and Management Server) of the TOE and securely re-transmitting the audit data to an external syslog server.

All communication between the NGFW engine and the SMC appliance occur over TLS protected communication channels. The NGFW engine authenticates its peer in these TOE-to-TOE communications using a customized certificate that is exchanged during the TLS negotiation.

The following communication pathways are represented in Figure 11.

- **Management server to Logging server communications** use the internal loopback interface within the SMC appliance. These communications involve the configuration of the Logging Server by the management Server.
- **Logging Server to External Syslog Server communications** use TLS to protect the audit data transmitted from the Logging Server to the external syslog server.
- **NGFW Engine to Logging server communications** use TLS to protect the audit data transmitted from the NGFW engine to the Logging server.
- **NGFW Engine to/from Management server communications** use TLS to protect the configuration information exchanged between the Management server and the NGFW engine.

Either party in this communication pathway can be configured to act as the client. Typically, the Management server initiates configuration changes by sending updated security policies to the NGFW engine. However, it is possible that such a communication pathway could be blocked, requiring the NGFW engine to initiate the configuration change. In this case, the NGFW engine is configured to poll for configuration changes on a regular basis (typically 10 minutes).

- **Client GUI to Management server communications** uses TLS to protect the communication over which remote administration actions occur.
- The **NGFW engine** controls connectivity and information flow between **internal and external connected networks**.

The cryptographic operations occurring as part of the communication on the SMC appliance involving the Management server and Logging server are performed using RSA's Crypto-J Library. This library provides the encryption, decryption, signing and hashing functions necessary to support the SMC appliance use of TLS to protect Internal-TOE-transfers, the trusted channel mechanism and the trusted path mechanism. The SMC appliance also uses the OpenSSL library that is included within the MLOS to perform signature verification supporting the TOE trusted update mechanism.

The NGFW engine utilizes the Engine OpenSSL Library to provide the encryption, decryption, signing and hashing functions necessary to support the NGFW engine's use of TLS to protect Internal-TOE-transfers.

1.4.1.1 Physical Boundaries

The TOE is composed of two physical components: the NGFW engine appliance and the SMC appliance. Each of these appliances have physical network connections to its environment to facilitate communication between TOE components as well as to position the TOE to monitor and filter network traffic. All management of the TOE occurs through the SMC appliance, while all firewall packet filtering occurs through the NGFW engine.

The TOE is accessed and managed from a PC in the environment which is expected to have a communication pathway to the SMC appliance.

The TOE can be configured to forward its audit records to an external syslog server in the environment. All audit records sent to the external syslog server, are sent from the SMC appliance. The NGFW engine does not send audit data directly to an external syslog server. Instead, the NGFW engine passes all of its audit data to the Logging server on the SMC appliance, which forwards the data to the external syslog server.

The TOE can be configured to synchronize its internal clock using an NTP server in the operational environment. The SMC appliance synchronizes with the external NTP server, then configures the NGFW engine's time to be in synch with itself. The NGFW engine does not synchronize to the external NTP server itself.

The NGFW engine utilizes the Engine OpenSSL Library to support the NGFW engine's use of TLS to protect Internal-TOE-transfers. The SMC appliance uses RSA's Crypto-J Library to provide TLS, which protects Internal-TOE-transfers, the trusted channel mechanism and the trusted path mechanism.

1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by Next Generation Firewall:

- Security audit
- Cryptographic support
- User data protection
- Stateful Traffic Filtering Firewall
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

1.4.1.2.1 Security audit

The TOE generates audit events for numerous activities including policy enforcement, system management and authentication. A syslog server in the environment is relied on to store audit records generated by the TOE. The TOE generates a complete audit record including the IP address of the TOE, the event details, and the time the event occurred. The time stamp is provided by the TOE appliance hardware. When the syslog server writes the audit record to the audit trail, it applies its own time stamp, placing the entire TOE-generated syslog protocol message MSG contents into an encapsulating syslog record.

1.4.1.2.2 Cryptographic support

Because the TOE is distributed into two physically distinct parts, each physical component of the TOE must be considered when discussing the TOE cryptographic support. Both components of the TOE utilize cryptography to support its use of the TLS protocol to protect network communication and to support verification of TOE updates.

1.4.1.2.3 User data protection

The TOE ensures that residual information is protected from potential reuse in accessible objects such as network packets.

1.4.1.2.4 Identification and authentication

The TOE requires users to be identified and authenticated before they can use functions mediated by the TOE, with the exception of reading the login banner, and performing firewall packet filtering operations. The TOE authenticates administrative users. In order for an administrative user to access the TOE, a user account including a user name and password must be created for the user.

1.4.1.2.5 Security management

Security management commands are limited to authorized users (i.e., administrators) and available only after they have provided acceptable user identification and authentication data to the TOE. Administrators access the TOE remotely using a TLS protected communication channel between the Management server and the Client GUI (which runs on a workstation in the IT environment).

1.4.1.2.6 Protection of the TSF

The TOE provides a variety of means of protecting itself. The TOE performs self-tests that cover the correct operation of the TOE. It provides functions necessary to securely update the TOE. It provides a hardware clock to ensure reliable timestamps. It protects sensitive data such as stored passwords and cryptographic keys so that they are not accessible through the TOE, even to an authorized administrator. The TOE also utilizes the TLS protocol to protect communication between distributed parts of the TOE.

1.4.1.2.7 TOE access

The TOE can be configured to display a logon banner before a user session is established. The TOE also enforces inactivity timeouts for local and remote sessions.

1.4.1.2.8 Trusted path/channels

The TOE protects interactive communication with administrators using TLS for GUI access, ensuring both integrity and disclosure protection. If the negotiation of an encrypted session fails the attempted connection will not be established.

The TOE protects communication with network peers, such as an external syslog server, using TLS connections to prevent unintended disclosure or modification of logs.

The TOE also protects internal communication between components of the TOE using TLS connections which prevent unintended disclosure and modification of TSF communications.

1.4.2 TOE Documentation

Forcepoint offers a series of documents that describe the installation process for the TOE as well as guidance for subsequent use and administration of the applicable security features. The following list of documents was examined as part of the evaluation.

- Intel Security Product Guide, McAfee Next Generation Firewall 5.10, Revision A
- Intel Security Installation Guide, McAfee Next Generation Firewall 5.10, Revision B
- McAfee Next Generation Firewall 5.10.1 Common Criteria Evaluated Configuration Guide, Revision F
- McAfee Security Management Center Appliance Hardware Guide, Revision B
- Hardware Guide, Revision D, McAfee Next Generation Firewall, Models 321, 325, 1035, 1065, 1401, 1402
- Hardware Guide, Revision D, McAfee Next Generation Firewall, Models 3201, 3202, 3205, 3206, 3207, 3301
- Hardware Guide, Revision B, McAfee Next Generation Firewall, Models 5201, 5205, 5206
- Hardware Guide, Revision B, McAfee Next Generation Firewall, Model 320X

2 Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 4, September 2012.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1 Revision 4, September 2012.
    - Part 3 Conformant
- Package Claims:
    - Protection Profile for Network Devices, Version 1.1 (with Errata #3), 8 June 2012 and Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Filter Firewall, Version 1.0, 19 December 2011 (NDPP11e3/STFFEP10)

2.1 Conformance Rationale

The ST conforms to the NDPP11e3/STFFEP10. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the PP.

3 Security Objectives

The Security Problem Definition may be found in the NDPP11e3/STFFEP10 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDPP11e3/STFFEP10 offers additional information about the identified security objectives, but that has not been reproduced here and the NDPP11e3/STFFEP10 should be consulted if there is interest in that material.

In general, the NDPP11e3/STFFEP10 has defined Security Objectives appropriate for network infrastructure device and as such are applicable to the Stonesoft Next Generation Firewall TOE.

3.1 Security Objectives for the Operational 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.PHYSICAL Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

OE.TRUSTED_ADMIN TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

OE.CONNECTIONS TOE administrators will ensure that the TOE is installed in a manner that will allow the TOE to effectively enforce its policies on network traffic flowing among attached networks.

4 Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDPP11e3/STFFEP10. The NDPP11e3/STFFEP10 defines the following extended requirements and since they are not redefined in this ST the NDPP11e3/STFFEP10 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- FAU_STG_EXT.1: External Audit Trail Storage
- FCS_CKM_EXT.4: Cryptographic Key Zeroization
- FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
- FFW_RUL_EXT.1: Stateful Traffic Filtering
- FIA_PMG_EXT.1: Password Management
- FIA_UAU_EXT.2: Extended: Password-based Authentication Mechanism
- FIA_UIA_EXT.1: User Identification and Authentication
- FPT_APW_EXT.1: Extended: Protection of Administrator Passwords
- FPT_SKP_EXT.1: Extended: Protection of TSF Data (for reading of all symmetric keys)
- FPT_TST_EXT.1: TSF Testing
- FPT_TUD_EXT.1: Extended: Trusted Update
- FTA_SSL_EXT.1: TSF-initiated Session Locking

5 Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the NDPP11e3/STFFEP10. The refinements and operations already performed in the NDPP11e3/STFFEP10 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDPP11e3/STFFEP10 and any residual operations have been completed herein. Of particular note, the NDPP11e3/STFFEP10 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the NDPP11e3/STFFEP10 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the NDPP11e3/STFFEP10 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The NDPP11e3/STFFEP10 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Stonesoft Next Generation Firewall TOE.

Table 5-1 TOE Security Functional Components

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User Identity Association
	FAU_STG_EXT.1: External Audit Trail Storage
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation (for asymmetric keys)
	FCS_CKM_EXT.4: Cryptographic Key Zeroization
	FCS_COP.1(1): Cryptographic Operation (for data encryption/decryption)
	FCS_COP.1(2): Cryptographic Operation (for cryptographic signature)
	FCS_COP.1(3): Cryptographic Operation (for cryptographic hashing)
	FCS_COP.1(4): Cryptographic Operation (for keyed-hash message authentication)
	FCS_RBG_EXT.1: Extended: Cryptographic Operation (Random Bit Generation)
	FCS_TLS_EXT.1: Explicit: TLS
FDP: User data protection	FDP_RIP.2: Full Residual Information Protection
FFW: Stateful Traffic Filtering Firewall	FFW_RUL_EXT.1: Stateful Traffic Filtering
FIA: Identification and authentication	FIA_PMG_EXT.1: Password Management
	FIA_UAU.7: Protected Authentication Feedback
	FIA_UAU_EXT.2: Extended: Password-based Authentication Mechanism
	FIA_UIA_EXT.1: User Identification and Authentication
FMT: Security management	FMT_MTD.1: Management of TSF Data (for general TSF data)
	FMT_SMF.1: Specification of Management Functions
	FMT_SMR.2: Restrictions on Security Roles
FPT: Protection of the TSF	FPT_APW_EXT.1: Extended: Protection of Administrator Passwords
	FPT_ITT.1: Basic Internal TSF Data Transfer Protection
	FPT_SKP_EXT.1: Extended: Protection of TSF Data (for reading of all symmetric keys)

	FPT_STM.1: Reliable Time Stamps
	FPT_TST_EXT.1: TSF Testing
	FPT_TUD_EXT.1: Extended: Trusted Update
FTA: TOE access	FTA_SSL.3: TSF-initiated Termination
	FTA_SSL.4: User-initiated Termination
	FTA_SSL_EXT.1: TSF-initiated Session Locking
	FTA_TAB.1: Default TOE Access Banners
FTP: Trusted path/channels	FTP_ITC.1: Inter-TSF trusted channel
	FTP_TRP.1: Trusted Path

5.1.1 Security audit (FAU)

5.1.1.1 Audit Data Generation (FAU_GEN.1)

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;
d) Specifically defined auditable events listed in Table 1 (in the NDPP11e3).
e) Specifically defined auditable events listed in Table 4-3 (in the STFFEP10).

Table 5-2 Audit Events

Requirement	Auditable Events	Additional Content
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG_EXT.1	None	None
FCS_CKM.1	None	None
FCS_CKM_EXT.4	None	None
FCS_COP.1(1)	None	None
FCS_COP.1(2)	None	None
FCS_COP.1(3)	None	None
FCS_COP.1(4)	None	None
FCS_RBG_EXT.1	None	None
FCS_TLS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
FDP_RIP.2	None	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
FIA_PMG_EXT.1	None	None
FIA_UAU.7	None	None
FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UIA_EXT.1	All use of the identification and	Provided user identity, origin of the

	authentication *mechanism*.	attempt (e.g., IP address).
FMT_MTD.1	None	None
FMT_SMF.1	None	None
FMT_SMR.2	None	None
FPT_APW_EXT.1	None	None
FPT_ITT.1	None	None
FPT_SKP_EXT.1	None	None
FPT_STM.1	Changes to the time.	The old and new values for the time. Origin of the attempt (e.g., IP address).
FPT_TST_EXT.1	None	None
FPT_TUD_EXT.1	Initiation of update.	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_SSL_EXT.1	Any attempts at unlocking 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	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.

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 PP/ST, information specified in column three of Table 1 (in the NDPP).

5.1.1.2 User Identity Association (FAU_GEN.2)

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.

5.1.1.3 External Audit Trail Storage (FAU_STG_EXT.1)

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 implementing the [***TLS***] protocol.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (for asymmetric keys) (FCS_CKM.1)

FCS_CKM.1.1

Refinement: The TSF shall generate asymmetric cryptographic keys used for key establishment in accordance with [

- ***NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' for elliptic curve-based key establishment schemes and implementing 'NIST curves' P-256, P-384 and [P-521] (as defined in FIPS PUB 186-3, 'Digital Signature Standard');***
- ***NIST Special Publication 800-56B, 'Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography' for RSA-based key establishment schemes***]

and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

5.1.2.2 Cryptographic Key Zeroization (FCS_CKM_EXT.4)

FCS_CKM_EXT.4.1

The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

5.1.2.3 Cryptographic Operation (for data encryption/decryption) (FCS_COP.1(1))

FCS_COP.1(1).1

Refinement: The TSF shall perform encryption and decryption in accordance with a specified cryptographic algorithm AES operating in [***CBC, GCM***] and cryptographic key sizes 128-bits and 256-bits that meets the following:

- FIPS PUB 197, 'Advanced Encryption Standard (AES)'
- [***NIST SP 800-38A, NIST SP 800-38D***].

5.1.2.4 Cryptographic Operation (for cryptographic signature) (FCS_COP.1(2))

FCS_COP.1(2).1

Refinement: The TSF shall perform cryptographic signature services in accordance with a [

(2) RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater, (3) Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater]

that meets the following:

[***Case: RSA Digital Signature Algorithm - FIPS PUB 186-2 or FIPS PUB 186-3, 'Digital Signature Standard', Case: Elliptic Curve Digital Signature Algorithm - FIPS PUB 186-3, 'Digital Signature Standard' - The TSF shall implement 'NIST curves' P-256, P-384 and [P-521] (as defined in FIPS PUB 186-3, 'Digital Signature Standard')***].

5.1.2.5 Cryptographic Operation (for cryptographic hashing) (FCS_COP.1(3))

FCS_COP.1(3).1

Refinement: The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384, SHA-512***] and message digest sizes [***160, 256, 384, 512***] bits that meet the following: FIPS Pub 180-3, 'Secure Hash Standard.'

5.1.2.6 Cryptographic Operation (for keyed-hash message authentication) (FCS_COP.1(4))

FCS_COP.1(4).1

Refinement: The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm HMAC-[***SHA-1, SHA-256, SHA-384, SHA-512***], key size [**assignment: 128, 160, 256, 384, 512**], and message digest sizes [***160, 256, 384, 512***] bits that meet the following: FIPS Pub 198-1, 'The Keyed-Hash Message Authentication Code', and FIPS Pub 180-3, 'Secure Hash Standard.'

5.1.2.7 Extended: Cryptographic Operation (Random Bit Generation) (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1

The TSF shall perform all random bit generation (RBG) services in accordance with [***NIST Special Publication 800-90 using [HMAC_DRBG (any), CTR_DRBG (AES)]***] seeded by an entropy source that accumulated entropy from [***a software-based noise source***].

FCS_RBG_EXT.1.2

The deterministic RBG shall be seeded with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

1.1.1.1 Explicit: TLS (FCS_TLS_EXT.1)

FCS_TLS_EXT.1.1

The TSF shall implement one or more of the following protocols [***TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:

Mandatory Ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:

- [***TLS_RSA_WITH_AES_256_CBC_SHA,
TLS_RSA_WITH_AES_128_CBC_SHA256,
TLS_RSA_WITH_AES_256_CBC_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384***].

5.1.3 User data protection (FDP)

5.1.3.1 Full Residual Information Protection (FDP_RIP.2)

FDP_RIP.2.1

The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] all objects.

5.1.4 Stateful Traffic Filtering Firewall (FFW)

5.1.4.1 Stateful Traffic Filtering (FFW_RUL_EXT.1)

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 process the following network traffic protocols:

- Internet Control Message Protocol version 4 (ICMPv4)
- Internet Control Message Protocol version 6 (ICMPv6)
- Internet Protocol (IPv4)
- Internet Protocol version 6 (IPv6)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

and be capable of inspecting network packet header fields defined by the following RFCs to the extent mandated in the other elements of this SFR

- RFC 792 (ICMPv4)
- RFC 4443 (ICMPv6)
- RFC 791 (IPv4)
- RFC 2460 (IPv6)
- RFC 793 (TCP)
- RFC 768 (UDP).

FFW_RUL_EXT.1.3

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
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

and distinct interface.

FFW_RUL_EXT.1.4

The TSF shall allow the following operations to be associated with Stateful Traffic Filtering rules: permit, deny, and log.

FFW_RUL_EXT.1.5

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

FFW_RUL_EXT.1.6

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, [***ICMP***] 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. [***ICMP: source and destination addresses, [type, code]'***].

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.1.7

The TSF shall be able to process the following network protocols:

1. FTP,
2. [***no other protocols***],

to dynamically define rules or establish sessions allowing network traffic of the following types:

- FTP: TCP data sessions in accordance with the FTP protocol as specified in RFC 959,
- [***no other protocols***].

FFW_RUL_EXT.1.8

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

1. The TSF shall reject and be capable of logging packets which are invalid fragments;
2. The TSF shall reject and be capable of logging fragmented IP packets which cannot be re-assembled completely;
3. The TSF shall reject 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;
4. The TSF shall reject 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;
5. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a broadcast network;
6. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a multicast network;
7. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
8. The TSF shall reject and be capable of logging network packets where the source address of the network packet is a multicast;
9. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is a link-local address;
10. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being an address 'reserved for future use' as specified in RFC 5735 for IPv4;
11. The TSF shall reject 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' as specified in RFC 3513 for IPv6;
12. The TSF shall reject and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
13. [***no other rules***].

FFW_RUL_EXT.1.9

When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall process the applicable Stateful Traffic Filtering rules (as determined in accordance with FFW_RUL_EXT.1.5) in the following order: administrator-defined.

FFW_RUL_EXT.1.10

When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall deny packet flow if a matching rule is not identified.

5.1.5 Identification and authentication (FIA)

5.1.5.1 Password Management (FIA_PMG_EXT.1)

FIA_PMG_EXT.1.1

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

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters: [***[, ! , @, #, $, %, ^, &, *, (,),]***];
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater.

5.1.5.2 Protected Authentication Feedback (FIA_UAU.7)

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.

5.1.5.3 Extended: Password-based Authentication Mechanism (FIA_UAU_EXT.2)

FIA_UAU_EXT.2.1

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

5.1.5.4 User Identification and Authentication (FIA_UIA_EXT.1)

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;
- [***[passing network traffic through the firewall engine]***].

FIA_UIA_EXT.1.2

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

5.1.6 Security management (FMT)

5.1.6.1 Management of TSF Data (for general TSF data) (FMT_MTD.1)

FMT_MTD.1.1

The TSF shall restrict the ability to manage the TSF data to the Security Administrators.

5.1.6.2 Specification of Management Functions (FMT_SMF.1)

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 update the TOE, and to verify the updates using [***digital signature***] capability prior to installing those updates; [
- ***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 configure the cryptographic functionality***
- ***Ability to configure firewall rules***].

5.1.6.3 Restrictions on Security Roles (FMT_SMR.2)

FMT_SMR.2.1

The TSF shall maintain the roles: Authorized 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

- Authorized Administrator role shall be able to administer the TOE locally;
- Authorized Administrator role shall be able to administer the TOE remotely;

are satisfied

5.1.7 Protection of the TSF (FPT)

5.1.7.1 Extended: Protection of Administrator Passwords (FPT_APW_EXT.1)

FPT_APW_EXT.1.1

The TSF shall store passwords in non-plaintext form.

FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext passwords.

5.1.7.2 Basic Internal TSF Data Transfer Protection (FPT_ITT.1)

FPT_ITT.1.1

Refinement: The TSF shall protect TSF data from disclosure and detect its modification when it is transmitted between separate parts of the TOE through the use [***TLS***].

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

FPT_SKP_EXT.1.1

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

5.1.7.4 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1

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

5.1.7.5 TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1

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

5.1.7.6 Extended: Trusted Update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2

The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.

FPT_TUD_EXT.1.3

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

5.1.8 TOE access (FTA)

5.1.8.1 TSF-initiated Termination (FTA_SSL.3)

FTA_SSL.3.1

Refinement: The TSF shall terminate a remote interactive session after a Security Administrator-configurable time interval of session inactivity.

5.1.8.2 User-initiated Termination (FTA_SSL.4)

FTA_SSL.4.1

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

5.1.8.3 TSF-initiated Session Locking (FTA_SSL_EXT.1)

FTA_SSL_EXT.1.1

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

5.1.8.4 Default TOE Access Banners (FTA_TAB.1)

FTA_TAB.1.1

Refinement: 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.

5.1.9 Trusted path/channels (FTP)

5.1.9.1 Inter-TSF trusted channel (FTP_ITC.1)

FTP_ITC.1.1

Refinement: The TSF shall use [***TLS***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***[no other capabilities]***] 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 [***syslog***].

5.1.9.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1

Refinement: The TSF shall use [***TLS***] provide a trusted communication path between itself and 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 detection of modification of the communicated data.

FTP_TRP.1.2

Refinement: The TSF shall permit remote administrators to initiate communication via the trusted path.

FTP_TRP.1.3

The TSF shall require the use of the trusted path for initial administrator authentication and all remote administration actions.

5.2 TOE Security Assurance Requirements

The SARs for the TOE are the components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

Table 5-3 Assurance Components

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1: Basic functional specification
AGD: Guidance documents	AGD_OPE.1: Operational user guidance
	AGD_PRE.1: Preparative procedures
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM coverage
ATE: Tests	ATE_IND.1: Independent testing - conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability survey

5.2.1 Development (ADV)

5.2.1.1 Basic functional specification (ADV_FSP.1)

ADV_FSP.1.1d

The developer shall provide a functional specification.

ADV_FSP.1.2d

The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.1.1c

The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2c

The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c

The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4c

The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.1.1e

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

ADV_FSP.1.2e

The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational user guidance (AGD_OPE.1)

AGD_OPE.1.1d

The developer shall provide operational user guidance.

AGD_OPE.1.1c

The operational user guidance shall describe, for each user role, the user-accessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c

The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c

The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c

The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c

The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences and implications for maintaining secure operation.

AGD_OPE.1.6c

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfil the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

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

5.2.2.2 Preparative procedures (AGD_PRE.1)

AGD_PRE.1.1d

The developer shall provide the TOE including its preparative procedures.

AGD_PRE.1.1c

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1e

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

AGD_PRE.1.2e

The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d

The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1c

The TOE shall be labelled with its unique reference.

ALC_CMC.1.1e

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

5.2.3.2 TOE CM coverage (ALC_CMS.1)

ALC_CMS.1.1d

The developer shall provide a configuration list for the TOE.

ALC_CMS.1.1c

The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2c

The configuration list shall uniquely identify the configuration items.

ALC_CMS.1.1e

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

5.2.4 Tests (ATE)

5.2.4.1 Independent testing - conformance (ATE_IND.1)

ATE_IND.1.1d

The developer shall provide the TOE for testing.

ATE_IND.1.1c

The TOE shall be suitable for testing.

ATE_IND.1.1e

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

ATE_IND.1.2e

The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability survey (AVA_VAN.1)

AVA_VAN.1.1d

The developer shall provide the TOE for testing.

AVA_VAN.1.1c

The TOE shall be suitable for testing.

AVA_VAN.1.1e

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

AVA_VAN.1.2e

The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3e

The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6 TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- User data protection
- Stateful Traffic Filtering Firewall
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

6.1 Security audit

The TOE generates audit records for all events identified by the requirement. The TOE audit mechanism cannot be disabled. The records include the required data, time, type, subject, outcome and event type values.

The startup and shutdown of the audit function is synonymous with the start-up and shutdown of the TOE. The set of potential audit events and record information include all of the events defined in Table 6-1 NGFW Audited Events.

The audit mechanism associated with firewall rules is the “logging” operation which is triggered using the logging option of a rule in the TOE security policy. The TOE applies the matching mechanism for packet filtering, and for each match a logging option can be defined that generates an audit record. The TSF selects the audited events based on the defined logging options. In addition to the logging operation, the TOE provides an audit record when the TOE security policy (i.e., active file) changes. When the TOE receives a new security policy it generates an audit record identifying the date, time, and configuration identification. The components of the TOE rely on the component’s operating system to provide the time for the audit records. The Management Server generates audit records providing the details on the use of the security management functions. The NGFW engine generates audit events pertaining to packet filtering.

The NGFW engine transfers audit records to the SMC appliance logging manager immediately after generation of the record. The SMC appliance log manager sends audits to an external syslog server immediately after they have been received. The SMC appliance management server generates audit records, stores the records locally and sends them to an external syslog server immediately after storing the records. When a connection to the external syslog server fails, the management server or logging server will be re-established and NEW audit records are sent to the syslog server.

Table 6-1 NGFW Audited Events

Requirement	Auditable Events	Additional Content
FAU_GEN.1	Start & Stop of the Audit function	None
FCS_TLS_EXT.1	Failure to establish a TLS Session. Establishment/Termination of a TLS session.	Reason for failure. Non-TOE endpoint of connection (IP address) for both successes and failures.
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
FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UIA_EXT.1	All use of the identification and authentication *mechanism*.	Provided user identity, origin of the attempt (e.g., IP address).
FPT_STM.1	Changes to the time.	The old and new values for the time. Origin of the attempt (e.g., IP address).
FPT_TUD_EXT.1	Initiation of update.	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_SSL_EXT.1	Any attempts at unlocking of an interactive session.	None
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.

TOE audit entries are first stored on cache buffers on each appliance. The size of this cache depends on the size of the disk in the appliance. The proprietary protocol for synchronizing and managing the data among the distributed components notifies the Logging Server that there is new log information and sends the log entry to the Logging Server. The audit information is stored by the Logging Server as database files which are only accessible to a TOE administrator via the Management Server. An audit entry is removed from cache buffers after the NGFW engine has received confirmation from Logging Server that the entry has been successfully stored.

The administrator defines the log spool policy. This specifies the behavior of the TOE whenever its local log spool is filled. The TOE supports two settings, but requires that only the following be used when in an evaluated configuration:

- Stop traffic (required in the evaluated configuration): NGFW engine automatically goes to an offline state and connections going through NGFW engine are transferred to other nodes in a cluster. Once the spool situation has improved, the node returns automatically to online state.

The TOE also provides a means for the Management Server to prioritize log data. The mechanism is based on the following log level:

- Alert: generated with an alert status and are always stored;
- Essential: always generated even if the NGFW Engine is running out of disk space;
- Stored: stored to the audit log database if alert and essential log entries have already been stored;
- Transient: not stored to database but kept in TOE log cache.

Before applying the selected log spooling policy, the engine stops producing transient logs. If insufficient, it can drop all but the essential log entries. As a last resort, the engine applies the selected log spooling policy.

The only TLS protocol errors that are logged are the termination of a session due to user authentication failure.

The Security audit function is designed to satisfy the following security functional requirements:

- FAU_GEN.1: The set of potential audit events and record information include all of the events defined in the table above. The records include the required data, time, type, subject, outcome and event type values. The TOE also inserts into audit records all of the additional information described by the table above.
- FAU_GEN.2: Every audit record indicates the SMC user responsible for the action.
- FAU_STG_EXT.1: All audit records generated by the NGFW are transmitted to an external syslog server using a TLS protected communication channel.

6.2 Cryptographic support

The TOE utilizes cryptographic support features as part of the TLS protocol mechanism as well as to verify software (both TOE updates and installed software). Each component of the TOE utilizes the cryptographic module available to it as follows:

- NGFW engine uses the Engine OpenSSL Library (TLS & TOE Update) version 1.0.1p-fips
- SMC Appliance uses the MLOS OpenSSL Library (TOE Update only) version 1.0.1p-fips
- SMC Appliance’s Management Server uses RSA BSAFE® Crypto-J library (TLS) version 6.1.2
- SMC Appliance’s Logging Server uses RSA BSAFE® Crypto-J library (TLS) version 6.1.2

Table 6-2 Crypto modules and FIPS Certificates

Functions	TOE Component	Standards	CAVP Certificates
Encryption / Decryption			
AES 128/256 CBC, GCM mode	NGFW Engine – Engine OpenSSL Library	FIPS PUB 197 (AES) 128/256-bit keys NIST SP 800-38A, NIST SP 800-38D	Cert #3517 / 3518
	SMC Appliance Management Server & Logging Server – RSA BSAFE® Crypto-J library	FIPS PUB 197 (AES) NIST SP 800-38A, NIST SP 800-38D	Cert #2249
Cryptographic signature			
RSA 2048 ECDSA 256-bit using P-521 curve	NGFW Engine – Engine OpenSSL Library	RSA: FIPS PUB 186-2 or 186-3 ECDSA: FIPS PUB 186-3 with NIST P-521 curve	Cert #1806 Cert #717
RSA 2048 ECDSA 256-bit using P-256, P-384, and P-521 curves	SMC Appliance Management Server & Logging Server – RSA BSAFE® Crypto-J library	RSA: FIPS PUB 186-2 or 186-3 ECDSA: FIPS PUB 186-3 with NIST P-256, P-384 and P-521 curves	Cert #1154 Cert #357
ECDSA 256-bit using P-521 curve	SMC Appliance MLOS – MLOS OpenSSL Library	ECDSA: FIPS PUB 186-3 with NIST P-521 curve	Cert #841

Cryptographic Hashing			
SHA-256/384	NGFW Engine – Engine OpenSSL Library	FIPS Pub 198-1 and FIPS Pub 180-3	Cert #2899
SHA-1/256/384/512	SMC Appliance Management Server & Logging Server – RSA BSAFE® Crypto-J library	FIPS Pub 198-1 and FIPS Pub 180-4	Cert #1938
SHA-384	SMC Appliance MLOS – MLOS OpenSSL Library	FIPS Pub 198-1 and FIPS Pub 180-3	Cert #3205
Keyed-hash Message Authentication			
HMAC SHA-1/256/384/512	NGFW Engine – Engine OpenSSL Library	FIPS 198-1 & 180-3	Cert #2245
	SMC Appliance Management Server & Logging Server – RSA BSAFE® Crypto-J library	FIPS 198-1 & 180-4	Cert #1378
HMAC SHA256	SMC Appliance MLOS – MLOS OpenSSL Library	FIPS 198-1 & 180-4	Cert #2523
Deterministic Random Bit Generation			
AES-256 CTR_DRBG	NGFW Engine – Engine OpenSSL Library	FIPS SP 800-90B	Cert #878
SHA-256 HMAC_DRBG	SMC Appliance Management Server & Logging Server – RSA BSAFE® Crypto-J library	FIPS SP 800-90B	Cert #273

6.2.1 NGFW Engine

The NGFW Engine uses the OpenSSL library (see Table 6-2) for all cryptographic operations. This includes encryption and decryption of TLS packets along with all hashing and signature related operations. The NGFW engine TLS implementation supports either ECDSA algorithms with the following cipher suites:

- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

During the course of operation, NGFW Engine must establish a communication channel with the SMC Management Server and the SMC Logging server. For communication with the Logging Server, the NGFW Engine initiates the channel and offers the cipher suites mentioned above. The Logging Server chooses the cipher suite to be used for communication between these two TOE components.

Communication between the NGFW Engine and Management server can be initiated by either endpoint, depending upon configuration. The Management Server typically initiates the TLS channel to push configuration updates to the TOE. However, if the network configuration precludes the NGFW engine from receiving inbound connections, the NGFW engine can be configured to periodically query the Management server to see if configuration changes are pending.

Regardless of which TOE component initiates the connection, all TOE-to-TOE communication pathways involving communication between the NGFW engine and the SMC appliance use only the cipher suites configured by the SMC appliance. The SMC appliance always configures TOE components to use TLSv1.2 with the following cipher suites;

- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

The NGFW Engine utilizes the Engine OpenSSL Library within the NGFW engine for all cryptography involved in support of TLS. The NGFW Engine also verifies the integrity of software installed on the engine using the Engine OpenSSL Library.

The NGFW Engine uses a software noise source, uses the Linux kernel Random Number Generator (LKRNG) to make entropy available throughout the system, and uses an OpenSSL 1.0.1p FIPS 2.0.8 provided SP 800-90A AES-256 CTR DRBG for key generation. This DRBG instantiates itself with 384-bits of seeding material drawn from an intermediary pool that in turn draws 512 bits from /dev/random. This design allows the DRBG to securely generate keys of any size.

The NGFW utilizes custom constructed certificates generated by the SMC Management server, to authenticate each TOE component during the TLS session establishment negotiations. The TOE-to-TOE mutual authentication and custom constructed certificates is described in 6.2.2. The NGFW engine generates its own key for this custom constructed certificate using the SP 800-90A AES-256 CTR DRBG.

The following tables specifically identify the “should”, “should not”, and “shall not” conditions from the publication along with an indication of how the TOE’s Engine OpenSSL Library conforms to those conditions.

Table 6-3 NIST SP800-56A Conformance

NIST SP800-56A Section Reference	“should”, “should not”, or “shall not”	Implemented?	Rationale for deviation
5.4	should	Yes	Not applicable
5.5.1.1	should	Yes	Not applicable
5.5.2	should	Yes	Not applicable
5.6.2	should	Yes	Not applicable
5.6.2.1	should	Yes	Not applicable
5.6.2.2	should	Yes	Not applicable
5.6.2.3	should	Yes	Not applicable
5.6.3.1	should	Yes	Not applicable
5.6.3.2.1	should	Yes	Not applicable
5.6.4.1	shall not	No	Not applicable
5.6.4.2	shall not	No	Not applicable
5.6.4.2	should	Yes	Not applicable
5.6.4.3	should (first occurrence)	Yes	Not applicable

NIST SP800-56A Section Reference	"should", "should not", or "shall not"	Implemented?	Rationale for deviation
5.6.4.3	should (second occurrence)	Yes	Not applicable
5.8	shall not (first occurrence)	No	Not applicable
5.8	shall not (second occurrence)	No	Not applicable
6	should (first occurrence)	Yes	Not applicable
6	should (second occurrence)	Yes	Not applicable
7	shall not (first occurrence)	No	Not applicable
7	shall not (second occurrence)	No	Not applicable
9	shall not	No	Not applicable

Table 6-4 NIST SP800-56B Conformance

NIST SP800-56B Section Reference	"should", "should not", or "shall not"	Implemented?	Rationale for deviation
5.6	should	Yes	Not applicable
5.8	shall not	No	Not applicable
5.9	shall not (first occurrence)	No	Not applicable
5.9	shall not (second occurrence)	No	Not applicable
6.1	should not	No	Not applicable
6.1	should (first occurrence)	Yes	Not applicable
6.1	should (second occurrence)	Yes	Not applicable
6.1	should (third occurrence)	Yes	Not applicable
6.1	should (fourth occurrence)	Yes	Not applicable
6.1	shall not (first occurrence)	No	Not applicable
6.1	shall not (second occurrence)	No	Not applicable
6.2.3	should	Yes	Not applicable
6.5.1	should	Yes	Not applicable
6.5.2	should	Yes	Not applicable
6.5.2.1	should	Yes	Not applicable
6.6	shall not	No	Not applicable
7.1.2	should	Yes	Not applicable
7.2.1.3	should	Yes	Not applicable
7.2.1.3	should not	No	Not applicable
7.2.2.3	should (first occurrence)	Yes	Not applicable
7.2.2.3	should (second occurrence)	Yes	Not applicable
7.2.2.3	should (third occurrence)	Yes	Not applicable
7.2.2.3	should (fourth occurrence)	Yes	Not applicable
7.2.2.3	should not	No	Not applicable
7.2.2.3	shall not	No	Not applicable
7.2.3.3	should (first occurrence)	Yes	Not applicable
7.2.3.3	should (second occurrence)	Yes	Not applicable
7.2.3.3	should (third occurrence)	Yes	Not applicable
7.2.3.3	should (fourth occurrence)	Yes	Not applicable
7.2.3.3	should (fifth occurrence)	Yes	Not applicable
7.2.3.3	should not	No	Not applicable

NIST SP800-56B Section Reference	“should”, “should not”, or “shall not”	Implemented?	Rationale for deviation
8	should	Yes	Not applicable
8.3.2	should not	No	Not applicable

6.2.2 SMC Appliance

The primary functions of the SMC appliance are provided by the Management Server and the Logging Server (described in following sections). However, there is one feature addressing the security functional requirements outlined in section 5.1, TOE Security Functional Requirements above. This feature is the ability of the TOE to securely update itself. The McAfee Linux Operating System performs this feature, independent of the operation of the Management Server and the Logging Server. The operation of the TOE update feature is described in section 6.7. However, the cryptography associated with verifying the validity of the update is provided by the MLOS OpenSSL Library.

The MLOS OpenSSL Library does not generate keys, but instead uses cryptography only to verify signatures and hashes associated with TOE updates.

The SMC Appliance uses a software noise source as input to a kernel DRBG which, in turn, provides output to user space (through both /dev/random and /dev/urandom). The SMC Appliance uses /dev/random to instantiate its RSA BSAFE® Crypto-J library’s SHA-256 HMAC_DRBG and generate keys. The RSA BSAFE® Crypto-J library is used by the Management Server and Logging Server as described below.

Despite the SMC using a different crypto library implementation, Table 6-3 NIST SP800-56A Conformance and Table 6-4 NIST SP800-56B Conformance also reflect the “should”, “should not”, and “shall not” conditions from the publication along with an indication of how the TOE’s RSA BSAFE® Crypto-J library conforms to those conditions.

6.2.2.1 Management Server

The SMC appliance’s Management Server builds and signs, custom constructed certificates that are used by the management server and every other TOE component for mutual authentication within the TLS protocol's session negotiations. Once the Management server is initially installed, it creates a custom constructed certificate for itself that is based upon either RSA 2048 or ECDSA P-521. The Management server assigns a unique serial number for every TOE component and inserts that serial number as the subject alternate name in certificates created as each TOE component initializes. The Management server signs each custom constructed certificate that it provides to each TOE component. These certificates are used ONLY to authenticate TOE components to one another in the context of Inter-TOE-Transfers. These certificates are never used to authenticate to external entities (e.g., an external syslog server).

The Management Server utilizes the RSA BSAFE® Crypto-J library for all encryption, decryption, hashing and signature operations associated with support for the TLS protocol. The Inter-TOE-Transfers involving the SMC Management server are protected by the TLS protocol, using the cipher suites described in section 6.2.1.

The Management Server communicates directly with an external syslog server to transmit audit records which it generates. Management Server audit records are not sent through the Logging server, but instead are transmitted directly to an external syslog server. The Management Server communicates with the external syslog server using TLSv1.2 protocol and the cipher suites identified by Table 6-5

The RSA BSAFE® Crypto-J library used by the Management Server uses the same /dev/urandom to instantiate its RSA BSAFE® Crypto-J library’s SHA-256 HMAC_DRBG and generate keys that is described as part of the SMC Appliance in 6.2.2.

6.2.2.2 Logging Server

The SMC Appliance's Logging server communicates with the NGFW engine over TLSv1.2 protected communication channels. Despite being on the same appliance, the communications between the Logging server and the Management server occur over TLS with mutual authentication. Communication between the Logging server and the Management server are initiated by the Management server to transfer configuration data to the logging server.

The Logging server utilizes the Java Crypto-J library for all encryption, decryption, hashing and signature operations associated with support for the TLS protocol. The Inter-TOE-Transfers involving the Logging server are protected by the TLS protocol, using the cipher suites described in section 6.2.1.

Communication between the Logging server and external syslog servers will be initiated by the Logging server. All connections to an external syslog server are protected using the TLSv1.2 protocol[3]. The Logging server is capable of utilizing the following cipher suites to communicate to an external syslog server.

Table 6-5 Cipher suites to communicate with an External Syslog Server

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, TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384, TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256, TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

The RSA BSAFE® Crypto-J library used by the Logging server uses the same /dev/urandom to instantiate its RSA BSAFE® Crypto-J library's SHA-256 HMAC_DRBG and generate keys that is described as part of the SMC Appliance in 6.2.2.

6.2.3 Cryptographic Support Summary

The following table presents the crypto security parameters (CSPs), secret keys, and private keys provided by the TOE. The table also identifies when each CSP or key is cleared.

CSP or Key:	Stored in	Zeroized upon:	Zeroized by:
TLS Host RSA or ECDSA private key	On Disk	Command	Overwriting with zeros
TLS host RSA or ECDSA digital signature	On Disk	Command	Overwriting with zeros
TLS pre-master secret	In Memory	Handshake done	Overwriting with pseudo random data
TLS session key	In Memory	Close of session	Overwriting with pseudo random data

[3] The TOE supports TLSv1.0, TLSv1.1 and TLSv1.2 for communication with external syslog servers, however, guidance instructs that only TLSv1.2 be configured.

CSP or Key:	Stored in	Zeroized upon:	Zeroized by:
Passwords	On Disk	Command	Overwriting once with zeros

Table 6 CSPs and Keys

The Cryptographic support function is designed to satisfy the following security functional requirements:

- FCS_CKM.1: Each TOE component components support asymmetric key generation for key establishment as part of TLS as described in the section above. The following table details which components act as TLS clients and servers as well as which ones generate RSA, DH, or ECDH keys used during DHE_*, ECDHE_* and TLS_RSA_* TLS cipher suites.

Server Component	Client/Server/Both	DH key gen?	ECDH key gen?	RSA key gen?
NGFW Engine	Both	No	Yes	Yes
Management Server	Both	No	Yes	Yes
Logging Server	Both	No	Yes	Yes

  The TOE crypto modules generate asymmetric cryptographic keys in accordance with NIST SP 800-56A, and NIST SP 800-56B as described above.

- FCS_CKM_EXT.4: All TOE components clear keys (TLS) from memory after those keys are no longer needed.

- FCS_COP.1(1): The NGFW performs encryption and decryption using AES in either CBC or GCM mode, and key sizes of either 128 or 256 as described in section 6.2. The crypto modules providing the AES implementation and the corresponding FIPS certifications are identified in Table 6-2 Crypto modules and FIPS Certificates.

- FCS_COP.1(2): The TOE supports the use of rDSA with 2048 bit key sizes, and ECDSA with a key size of 256 bits or greater for cryptographic signatures. The crypto modules providing the cryptographic signature services are identified in Table 6-2.

- FCS_COP.1(3): The TOE supports cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512. The crypto modules providing the cryptographic hashing services are identified in Table 6-2.

- FCS_COP.1(4): The TOE supports keyed-hash message authentication using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512. The crypto modules providing the keyed-hash message authentication with the corresponding FIPS certificates are identified in Table 6-2. Keyed Hashing is used for the following purposes with these key sizes:

  - TLS 1.2 master secret 384 bits,
  - RSA premaster secret 384 bits,
  - ECDHE premaster secret sizes for 256, 384 and 512 bits for P-256, P-384 and P-521, respectively.
  - NGFW Engine file system integrity check key size 128 bits using OpenSSL HMAC-SHA-256, and
  - TLS 1.2 will use key sizes 160, 256 and 384 bits for HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-384, respectively.

- FCS_RBG_EXT.1: The TOE components perform random bit generation as described in section 6.2.1, “NGFW Engine” and section 6.2.2, “SMC Appliance”.

6.3 User data protection

The TOE has been designed to ensure that no residual information exists in network packets. When the TOE allocates a new buffer for either an incoming or outgoing a network packet, the new packet data will be used to overwrite any previous data in the buffer. If an allocated buffer exceeds the size of the packet, and additional space will be overwritten (padded) with zeros before the packet is forwarded (to the external network or delivered to the appropriate, internal application).

The User data protection function is designed to satisfy the following security functional requirements:

- FDP_RIP.2: The TOE ensures that previous information contents of resources used for new objects are not discernible in any new object, such as network packets, as described above.

6.4 Stateful Traffic Filtering Firewall

The TOE provides an information flow control mechanism using a rule base that comprises a set of security policy rules, i.e., the firewall security policy. The NGFW engine enforces the firewall security policy on all traffic that passes through the engine, via its internal or external network Ethernet interfaces. The traffic is TCP, UDP, ICMPv4, ICPv6, connections over IPv4 and IPv6. The NGFW engine inspects and filters these protocols based upon their header fields as defined by their corresponding RFC (See Table 6-7).

The NGFW engine only permits traffic to pass through that has been explicitly allowed by the firewall security policy, and implements packet defragmentation to enforce the policy on entire IP packets. Administrators using the Management Server define the firewall security policy rules.

Table 6-7 Protocols & Fields Filtered by the TOE

Protocol	Related RFC[4]	Fields Inspected
ICMPv4	RFC 792	Type, Code
ICMPv6	RFC 4443	Type, Code
IPv4	RFC 791	Source Address, Destination Address, Transport layer protocol
IPv6	RFC 2460	Source Address, Destination Address, Transport layer protocol
TCP	RFC 793	Source Port, Destination Port
UDP	RFC 768	Source Port, Destination Port

Any network traffic passed by the NGFW engine must be explicitly allowed by a firewall rule or be part of an established session allowed by a rule, or it is dropped. This is true even in the case of attempts to flood a TOE interface (in which case some packets may be dropped, but are never passed violating policy).

The NGFW engine has been designed to ensure that no residual information exists in network packets. When the NGFW engine allocates a new buffer for either an incoming or outgoing a network packet, the new packet data will be used to overwrite any previous data in the buffer. If an allocated buffer exceeds the size of the packet, and additional space will be overwritten (padded) with zeros before the packet is forwarded (to the external network or delivered to the appropriate, internal application). The NGFW engine implements connection tracking to manage

[4] Compliance with these RFCs is demonstrated by in-house compliance testing.

the information flow control decisions for connections (i.e., stateful sessions) rather than packets, providing increased performance and support for firewall features that require packet information above the IP level (e.g., ICMP, TCP, UDP). The connection tracking mechanism stores the state information of each connection to allow packets belonging to an established connection to pass. The connection tracking uses the fields shown in the following table when determining whether a packet matches an allowed established session for the corresponding protocol. Connection tracking will eliminate existing connections immediately, upon completion of the flow or upon an inactivity timeout for the session.

Table 6-8 Connection Tracking Fields

Protocol	Connection Tracking
TCP	Source & Destination Address, Source & Destination Port, Sequence Number, Flags
UDP	Source & destination address, source & destination port
ICMP	Source and destination address, type, code.
FTP	TCP data session attributes

Connection tracking works closely with the protocol agents to manage the information flow control decisions based on information attributes at the different networking layers through the application layer to decide whether a packet should be granted access or not. The following protocol agents and their security function are within the scope of the evaluation: FTP (RFC 959).

The FTP Protocol Agent keeps track of the ports used in File Transfer Protocol (FTP) sessions. An FTP session starts with a control connection (by default, TCP port 21), and the communications continue using a dynamically allocated port. The FTP Protocol Agent opens the actual ports used in FTP sessions as needed so that the whole range of possible dynamic ports does not need to be allowed in the policy.

The NGFW engine follows a specific orderly algorithm to traverse the rule base for matching and filtering the traffic between its internal and external networks. Any traffic that is not explicitly accepted by the security policy is rejected by the firewall. The structure of the rule base and the capabilities of its associated protocol agents enable the TSF to make the information flow control decisions.

Each rule comprises matching criteria and target actions. If the matching criteria is verified (i.e., a comparison matches) the NGFW engine applies the target actions. Possible target actions include Allow, Discard and Refuse[5.] Access rules with the logging option, can create a log or alert entry each time they match. The logging option is in addition to the target action of a rule.

The NGFW engine compares the information attributes defined in Table 6-7 Protocols & Fields Filtered by the TOE with the matching criteria of the rule to determine whether to apply the rule. If applied the target actions are implemented and the additional capabilities and flow control rules defined in Table 6-9 Additional Stateful Filtering Rules are applied.

Table 6-9 Additional Stateful Filtering Rules

1) The NGFW engine rejects and can log packets which are invalid fragments;
2) The NGFW engine rejects and can log fragmented IP packets which cannot be re-assembled completely;
3) The NGFW engine rejects and can log 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;

[5] Additional target actions are supported such as “Continue” and “Jump” which support complex security policies.

4) The NGFW engine rejects and can log 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;
5) The NGFW engine rejects and can log network packets where the source address of the network packet is defined as being on a broadcast network;
6) The NGFW engine rejects and can log network packets where the source address of the network packet is defined as being on a multicast network;
7) The NGFW engine rejects and can log network packets where the source address of the network packet is defined as being a loopback address;
8) The NGFW engine rejects and can log network packets where the source address of the network packet is a multicast;
9) The NGFW engine rejects and can log network packets where the source or destination address of the network packet is a link-local address;
10) The NGFW engine rejects and can log network packets where the source or destination address of the network packet is defined as being an address 'reserved for future use' as specified in RFC 5735 for IPv4;
11) The NGFW engine rejects and can log 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' as specified in RFC 3513 for IPv6;
12) The NGFW engine rejects and can log network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified

The rule base is read from top down, and when the first matching rule is encountered the search stops and the TOE executes the matching rule. There are two exceptions to this:

a) Jump rule - this makes the search jump to a sub-rule base if the jump rule matches. The search will continue inside the sub-rule base until it either finds a matching rule or comes back empty-handed from the sub-rule base and continues searching through the main rule base;

b) Continue rule - when it matches, it will set some variables and then the search continues.

The NGFW Engine obtains time values from the local hardware clock when making the security policy decisions associated with time-based information flows.

During the NGFW Engine boot process, there is a lag between the time when the network interface is operational, and the time that the Stateful Traffic Filtering functionality is fully functioning. During this time, traffic flow through the appliance is disabled; and traffic to and from the appliance is controlled by a Default Filter that drops all external traffic to the appliance.

The Stateful Traffic Filtering Firewall function is designed to satisfy the following security functional requirements:

- FFW_RUL_EXT.1: The NGFW engine filters network traffic using a rule base that comprises a set of security policy rules. These rules allow for complex security policies to be defined which control the flow of network traffic through the NGFW engine. Controlled network traffic includes at least IPv4, IPv6, ICMPv4, ICMPv6, TCP and UDP protocols. Additional features of the firewall functionality are described above in section 6.4 Stateful Traffic Filtering Firewall

  The rule base is read from top down, and when the first matching rule is encountered the search stops and the TOE executes the matching rule. Any traffic that is not explicitly accepted by the security policy is rejected by the firewall.

6.5 Identification and authentication

The TOE authenticates administrative users by means of a local password mechanism. Passwords can be composed of upper or lower case letters, numbers, and special characters including “!”, “@”, “#”, “$”, %”, “^”, “&”, “*”, “[“,

"]", "(" and ")". Administrators can specify a minimum length for passwords, and passwords can be greater than 15 characters.

Prior to login, the TOE displays a warning banner on both the GUI and local console interface. The TOE supports the filtering and forwarding of network traffic through the NGFW engine prior to an administrative user being authenticated. The TOE requires login prior to allowing any TOE configuration actions.

The Identification and authentication function is designed to satisfy the following security functional requirements:

- FIA_PMG_EXT.1: Password for local accounts can be composed of upper or lower case letters, numbers, and special characters as described above. Administrators can specify minimum lengths for passwords, and passwords can be greater than 15 characters.
- FIA_UAU.7: All passwords entered by administrators are obscured when entered.
- FIA_UAU_EXT.2: The TOE authenticates administrative users by means of a local password mechanism.
- FIA_UIA_EXT.1: The TOE displays a banner, and filters network traffic prior to administrative login. The TOE also requires login prior to all administrative actions. The SMC Management Server only accepts TLSv1.2 connections for management operations.

6.6 Security management

The administrator's interface to the TOE through the GUI is presented by a Java program provided by Forcepoint Security.

Local administration using a command line interface provides a limited capability (trusted TOE update only), The remote administrative capability provides the majority of the administrative functions. Using the GUI, the administrator can configure the cryptographic functionality of the TOE. The administrator can also configure the firewall access rule base. Once configured, the network traffic flow controls enforced by the TOE are enforced without intervention by an administrator. The Management Client software (GUI) must be installed from a Forcepoint provided installer.

The Security management function is designed to satisfy the following security functional requirements:

- FMT_MTD.1: The TOE ensures that only security administrators can login and configure TOE services.
  - The TSF shall restrict the ability to manage the TSF data to the Security Administrators.
- FMT_SMF.1: Administrators can configure operation of the TOE through a GUI, including configuring cryptographic functionality and services available prior to login. The local command line interface is used by administrators to verify and install TOE updates (see section 6.7 Protection of the TSF).
- FMT_SMR.2: The TOE maintains an administrative role for users. Users in this role can perform administrative actions locally or remotely.

6.7 Protection of the TSF

The Management Server stores passwords with other configuration data in a database and synchronizes this database with the Linux password database (i.e., /etc/shadow....). Synchronization takes the form of the contents of the database overwriting the contents of the Linux password database. There is no administrative interface to view or manipulate the raw configuration database. The only interface to the database is through administrative actions which modify the contents of the database in a controlled manner. Passwords are hashed using SHA-512 when stored.

The communication between the NGFW Engine and the servers running on the SMC appliance are all protected using mutually authenticated sessions over the TLSv1.2 protocol. The communication pathways include:

a) NGFW engine to Logging server for transmission of audit data;

b) Management server and NGFW engine communicate to transfer configuration data and product updates; and

c) Management server to Logging Server for transfer of configuration data specific to the logging server.

Custom constructed certificates are created by the management server for each part of the TOE. These custom constructed certificates are distributed to the various parts of the TOE during the TOE installation process.

For the first communication pathway, the NGFW engine initiates the establishment of a TLS connection to the logging server. The NGFW engine is the TLS client for this communication. For every TLS channel established between distributed TOE components, peer authentication is performed by verifying the certificates exchanged as part of the TLS negotiation. Each TOE component ensures that the certificate presented by the peer was generated by the TOE's management server and that the peer's internal identity is in the certificate's subject alternate name field.

None of the TOE components utilize pre-shared keys or long-lived symmetric keys. The only keys retained by the components of the TOE are associated with certificates used for TLS. The servers running on the SMC appliance store private keys in a password protected java keystore. The NGFW engine stores its private key (associated with its custom constructed TLS certificate) in a Read/Write partition with other configuration data (/data/config/tls/private-keys.pem). Since there is no admin interface on the FWE, there is no way for an administrator to view private keys.

Every appliance that is included as part of the TOE (i.e., NGFW engines and SMC appliance) includes its own real-time hardware-based clock. The time values from this clock are used in audit records. The NGFW engines receives its time updates from the management server only. The Management server is responsible for accepting and propagating clock updates initiated by an administrator or by NTP. The NGFW engine does not utilize NTP for time changes.

Each component of the TOE includes a set of hardware validation tests which include Known Answer Tests (KAT) for the cryptographic features provided by the OpenSSL and RSA Crypto-J cryptographic modules. These KAT tests cover operation of AES, RSA, ECDSA SHA-512 and HMAC-SHA. For each KAT test, the TOE uses known data as inputs into each cryptographic function, computes a cryptographic result (e.g., the AES ciphertext or SHA-512 hash), and compares the calculated result to the expected/known value. If the two do not match, the TOE will halt its boot as a result of the error. The SMC Appliance also validates a HMAC-SHA-256 checksum of the TOE binaries upon system startup. The NGFW Engine uses the Engine OpenSSL Library (HMAC-SHA-256) with a hardcoded key to verify the hash of the whole partition containing TOE binaries (/data/config/base/rootfs.hmac-sha256).

The TOE performs trusted updates for both of its components: the SMC management appliance and NGFW engines. To update the TSF software of the NGFW engine, an administrator can either obtain an update from Forcepoint (and then upload the update to the SMC) or the administrator can configure the SMC to automatically download updates from Forcepoint (assuming that the SMC has internet connectivity). After the SMC has the update, the SMC will verify the Forcepoint ECDSA P-521 w/ SHA-512 signature on the update package and only if the signature verifies correctly, the SMC will import that package, making it available to update administrator specified NGFW engines with the new software. The NGFW engine, relying upon the SMC to have verified the ECDSA signature on the update, will use that update package (which are an rdiff image) to write to an internal, alternate software/system partition, and then after verifying the checksum of the newly written system partition to check for write corruptions, will reboot into that new partition.

To update the SMC itself, the administrator obtains an SMC patch and makes that available to the SMC (either through a USB thumb-drive or by uploading it to the SMC). Then using the local console Command Line Interface (CLI), the administrator executes the `ambr_load` function to verify a Forcepoint ECDSA P-521 w/ SHA-512 signature on the patch file. If the signature verifies, then the administrator can issue the `ambr_install` command to install the patch, and then the administrator can follow the installation process (which can require a reboot for upgrades or major new features).

The Protection of the TSF function is designed to satisfy the following security functional requirements:

- FPT_APW_EXT.1: Passwords are stored only on the SMC Appliance in a configuration database used by the Management Server and in the Linux password database. Both locations store passwords in a non-plaintext form. No interfaces are provided by the TOE to allow passwords to be viewed in plaintext form.
- FPT_ITT.1: All communication between distributed parts of the TOE is performed using a TLSv1.2 protected communication channel. Distributed parts of the TOE authenticate each other using custom constructed certificates which include unique serial numbers as part of the certificates.
- FPT_SKP_EXT.1: None of the TOE components utilize pre-shared keys or long-lived symmetric keys. The only keys retained by the components of the TOE are associated with certificates used for TLS. These keys are stored in password protected JAVA keystore (on the SMC Appliance) and on a RW partition on the NGFW engine. Since the NGFW engine does not support an interface for local administration, this data is not accessible once stored in the partition.
- FPT_STM.1: Each TOE component includes a hardware-based real-time clock. This clock is used for timestamps used in audit data and measuring session timeouts. The TOE time can be set by administrator action or using an external NTP server communicating with the SMC Management Server. Time on the NGFW Engine is updated by the Management Server only.
- FPT_TST_EXT.1: The TOE components verify memory operation and checksums of TOE binaries upon startup as described above.
- FPT_TUD_EXT.1: The administrator can query the current software versions for the SMC software and for the NGFW engine software. Administrators can obtain TOE patches either directly from Forcepoint or by configuring the SMC management server to automatically download patches. Administrators must initiate the installation of patches to the NGFW engine and to the SMC appliance. Patches include signatures to verify the validity of the new software. If the signature on an update cannot be verified, the update cannot be uploaded into the appliance.

6.8 TOE access

The GUI offered by the SMC appliance has a configurable banner that is displayed before a user's login. The banner contents are defined by the administrator through the GUI interface. This same banner is also displayed on the SMC appliance local console CLI prior to a user's login.

The SMC management server supports timeouts caused by inactivity through the GUI, as well as voluntary termination of a session (i.e., logout). When an administrator uses the local console's Command Line Interface (CLI), the CLI enforces an inactivity timeout value that terminates the session after the administrator-specified time period.

The TOE access function is designed to satisfy the following security functional requirements:

- FTA_SSL.3: The TOE will terminate remote interactive sessions that have been inactive for the defined interval. The administrator can configure the duration of the inactivity timeout mechanism.
- FTA_SSL.4: Administrators using the GUI or local console (i.e., CLI) can terminate their own session using the logoff commands provided by these interfaces.

- FTA_SSL_EXT.1: The only local interactive sessions are those offered by the SMC appliance providing a command line interface.
- FTA_TAB.1: A Banner is displayed on both interfaces offered by the SMC appliance (i.e., the GUI and local console). The NGFW engine does not offer a direct network interface.

6.9 Trusted path/channels

The only communication that the NGFW has with a trusted external IT entity is the syslog channel. This channel is protected by TLS. For this communication channel, the TOE is acting as the TLS client during the negotiation of the TLS connection. The TOE supports the use of a certificate provided by the syslog server, as the mechanism to authenticate the syslog server to the TOE. The TOE does not utilize the internal, custom built certificates that authenticate itself to other TOE components (i.e., the FPT_ITT.1 communications) when authenticating itself to the syslog server.

The administrator's interface to the TOE through the GUI is presented by a Java program provided by Forcepoint. This GUI is installed from a Forcepoint provided CDROM. The GUI interacts with the Management Server which performs all identification, authentication, and permission enforcement. The java program provides the graphical user interface only. All decisions on whether the operation is allowed occur in the Management Server. The Management Server only accepts TLS connections for management operations. The following are the cipher suites which the Management Server accepts when communicating with the GUI.

- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

The Trusted path/channels function is designed to satisfy the following security functional requirements:

- FTP_ITC.1: The TOE protects syslog communication from the logging server and from the management server to the external syslog server using the TLSv1.2 protocol.
- FTP_TRP.1: The SMC Management Server only accepts TLSv1.2 connections for management operations using the cipher suites mentioned immediately above.