Forcepoint NGFW 6.3.1 (FWcPP10) Security Target

Version 1.0
Mar 05, 2018

Prepared for:

Forcepoint

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

www.Forcepoint.com

Prepared By:

www.gossamersec.com

1. SECURITY TARGET INTRODUCTION 4

1.1 SECURITY TARGET REFERENCE 4
1.2 TOE REFERENCE 4
1.3 TOE OVERVIEW 5
1.4 TOE DESCRIPTION 5
1.4.1 TOE Architecture 5
1.4.2 TOE Documentation 9

2. CONFORMANCE CLAIMS 10

2.1 CONFORMANCE RATIONALE 10

3. SECURITY OBJECTIVES 11

3.1 SECURITY OBJECTIVES FOR THE OPERATIONAL ENVIRONMENT 11

4. EXTENDED COMPONENTS DEFINITION 12

5. SECURITY REQUIREMENTS 13

5.1 TOE SECURITY FUNCTIONAL REQUIREMENTS 13
5.1.1 Security audit (FAU) 14
5.1.2 Cryptographic support (FCS) 16
5.1.3 User data protection (FDP) 18
5.1.4 Firewall (FFW) 18
5.1.5 Identification and authentication (FIA) 20
5.1.6 Security management (FMT) 21
5.1.7 Protection of the TSF (FPT) 22
5.1.8 TOE access (FTA) 23
5.1.9 Trusted path/channels (FTP) 23
5.2 TOE SECURITY ASSURANCE REQUIREMENTS 24
5.2.1 Development (ADV) 24
5.2.2 Guidance documents (AGD) 24
5.2.3 Life-cycle support (ALC) 25
5.2.4 Tests (ATE) 26
5.2.5 Vulnerability assessment (AVA) 26

6. TOE SUMMARY SPECIFICATION 27

6.1 SECURITY AUDIT 27
6.2 CRYPTOGRAPHIC SUPPORT 28
6.2.1 NGFW Engine 29
6.2.2 SMC Appliance 29
6.2.3 Cryptographic Support Summary 31
6.3 USER DATA PROTECTION 32
6.4 FIREWALL 33
6.5 IDENTIFICATION AND AUTHENTICATION 35
6.6 SECURITY MANAGEMENT 36
6.7 PROTECTION OF THE TSF 37
6.8 TOE ACCESS 38
6.9 TRUSTED PATH/CHANNELS 39

LIST OF TABLES

Table 1 TOE Security Functional Components 14
Table 2 Audit events 15
Table 3 Assurance Components 24
Table 6-1 NGFW Engine Forcepoint NGFW FIPS Object Module CAVP Certificates 29
Table 6-2 SMC Appliance Forcepoint NGFW FIPS Object Module CAVP Certificates 29

Table 6-3 SMC Appliance Forcepoint NGFW FIPS Java API CAVP Certificates 29
Table 6-4 Cipher suites to communicate with an External Syslog Server 30
Table 6-5 Cipher suites to communicate with remote administrators 31
Table 6-6 CSPs and Keys 31
Table 6-7 Protocols & Fields Filtered by the TOE 33
Table 6-8 Connection Tracking Fields 34
Table 6-9 Additional Stateful Filtering Rules 35

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 NGFW 6.3.1 provided by Forcepoint. The TOE is being evaluated as a firewall

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 parenthetical number placed at the end of the component. For example FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement.
  - 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 …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.1 Security Target Reference

ST Title – Forcepoint NGFW 6.3.1 Security Target

ST Version – Version 1.0

ST Date – Mar 05, 2018

1.2 TOE Reference

TOE Identification –Forcepoint NGFW 6.3.1 which consists of:

Forcepoint NGFW Security Management Center (SMC) Appliance running software version 6.3.1:

Appliance

Forcepoint NGFW Engine running software version 6.3.1 and including the following models:

1U models: 1101, 1105, 1402, 2101, 2105,

2U models: 3301, 3305

4U model: 6205

TOE Developer – Forcepoint

Evaluation Sponsor – Forcepoint

1.3 TOE Overview

The Target of Evaluation (TOE) is the Forcepoint NGFW 6.3.1.

The Forcepoint 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 Forcepoint Next Generation Firewall is a stateful packet filtering firewall. The Forcepoint 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 NGFW Engines. Throughout the remainder of this document, references to the NGFW Engine are meant to reference the TOE's firewall engine, 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 contains a hardened Linux operating system (with a 4.9 kernel) executing on a single or multi-processor Forcepoint hardware platform.

The SMC Appliance (or SMC) contains the Management Server and Log Server. Like the NGFW Engine, the SMC contains a hardened Linux-based operating system (which uses a 2.6.32 kernel) to support the management capabilities and allow for the operation and configuration of firewall engines.

1.4.1 TOE Architecture

The Forcepoint Next Generation Firewall (NGFW) system is composed of two physical appliances: the NGFW Engine and the Security Management Center (SMC) Appliance. The NGFW Engine provides firewall functionality utilizing its Linux operating system and using its embedded Forcepoint NGFW FIPS Object Module library to provide all cryptographic functionality. The SMC Appliance provides Management Server and Log Server functionality. As the SMC utilizes both Java and C, the SMC relies upon both Forcepoint NGFW FIPS Java API with a Java runtime environment and Forcepoint NGFW FIPS Object Module for cryptographic functionality. In the evaluated configuration, the SMC Appliance must be directly connected to the NGFW Engine through a dedicated, local network connection.

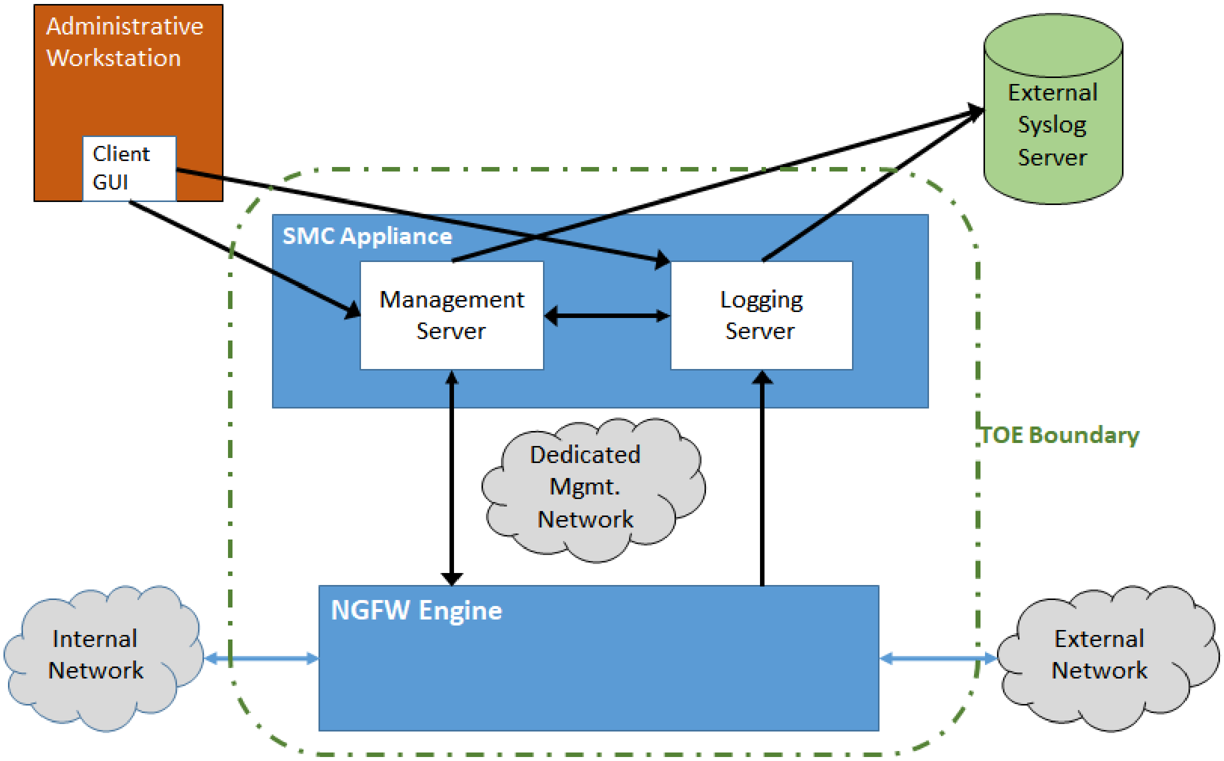

Figure 1 TOE Components, Communication Paths and IT Environment.

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 portion of the SMC Appliance provides the majority of the administrative capabilities in the NGFW system through the SMC Client GUI. The SMC Appliance provides a very limited console interface that allows administrators to verify and update TOE software, to manually set the time, and configure the console timeout.

The NGFW Engine does not have a local administrative interface, and can be configured through the SMC Appliance only. 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 the NGFW Engine components of the TOE and securely re-transmitting the audit data to an external syslog server. The Management Server component directly transmits its audit data to an external syslog server.

The administrator interfaces with the TOE mainly through the Client GUI, a Java program provided by Forcepoint. The administrator may download the GUI from the SMC Appliance using the Java Web Start or alternatively install it from a Forcepoint provided installation package. The Client GUI interacts with the Management Server which performs all identification, authentication, and permission enforcement. The Client GUI can also interact with the Log Server, allowing the administrator to query the NGFW Engine audit records that the Log Server has aggregated.

All communication between the NGFW Engine and the SMC Appliance occur over the dedicated, local network.

The following communication pathways are represented in **Figure 1 TOE Components, Communication Paths and IT Environment.**

- **Management Server to Log Server communications** use the internal loopback interface within the SMC Appliance. These communications involve the configuration of the Log Server by the management Server.
- **Management and Log Server to External Syslog Server communications** use TLS to protect the audit data transmitted from the Management and Log Server to the external syslog server.
- **NGFW Engine to Log Server communications** use the dedicated network to protect the audit data transmitted from the NGFW Engine to the Log Server.
- **NGFW Engine to/from Management Server communications** use the dedicated network to protect the configuration information exchanged between the Management Server and the NGFW Engine. Either party in this communication pathway can initiate the communications. Typically, the Management Server initiates configuration changes by sending updated security policies to the NGFW Engine. However, the NGFW Engine also polls for configuration changes on a regular basis.
- **Client GUI to Management and Log 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** that it is protecting.

The cryptographic operations occurring as part of the communication on the SMC Appliance involving the Management Server and Log Server are performed using the Forcepoint NGFW FIPS Java API (library). This provider provides the encryption, decryption, signing and hashing functions necessary to support the SMC Appliance use of the trusted channel mechanism and the trusted path mechanism. The SMC Appliance also uses the OpenSSL library to perform signature verification supporting the TOE trusted update mechanism.

The NGFW Engine utilizes its OpenSSL library to provide the encryption, decryption, signing and hashing functions necessary to support the NGFW Engine’s trusted update mechanism.

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, one dedicated, local network connection to exclusively facilitate communication between the SMC and engine, as well as additional network connections that positions the NGFW Engine portion of the TOE to monitor and filter network traffic. The SMC Appliance provides all management functionality, while the NGFW Engine provides all firewall packet filtering.

The TOE is accessed and managed from the Forcepoint Security Management Center Client (6.3.1) installed on a PC in the environment, where the PC is expected to have a network 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, a NGFW Engine passes all of its audit data to the Log Server on the SMC Appliance, which can (if configured) forward 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 directly with an external NTP server, but instead synchronizes only with the SMC.

The NGFW Engine utilizes its OpenSSL Library to verify trusted engine software updates. The SMC Appliance uses its Forcepoint NGFW FIPS Java API Library to provide TLS (which protects the trusted channel mechanism and the trusted path mechanism) and uses its OpenSSL library to verify SMC updates.

Each Engine model provides different performance as described in the table below.

Model	Form factor/CPU	Fixed ports	1G copper	10G Fiber	40G Fiber	Network I/O slots	Max FW throughput
1101	1U Pentium D1508	8x GE RJ45, 2x 10Gbps SFP+	8 to 16	2 to 6	0	1	50 Gbps
1105	1U Xeon D-1518	8x GE RJ45, 2x 10Gbps SFP+	8 to 16	2 to 6	0	1	60 Gbps
1402	1U Xeon E5-1650 v2	4x GE RJ45	4 to 20	0 to 8	0 to 4	2	40 Gbps
2101	1U Xeon D-1548	12x GE RJ45, 2x 10Gbps SFP+	12 to 28	2 to 10	0 to 4	2	60 Gbps
2105	1U Xeon D-1567	12x GE RJ45, 2x 10Gbps SFP+	12 to 28	2 to 10	0 to 4	2	80 Gbps
3301	2U Xeon E5-2618L v3	2x GE RJ45	2 to 34	0 to 16	0 to 8	4	80 Gbps
3305	2U Xeon E5-2680 v3	2x GE RJ45, 1x 40Gbps QSFP+	2 to 34	0 to 16	1 to 9	4	160 Gbps
6205	4U Xeon E5-2680 v4	2x GE RJ45	2 to 66	0 to 32	1 to 17	8	240 Gbps

1.4.1.2 Logical Boundaries

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

- Security audit
- Cryptographic support
- User data protection
- 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’s Linux-based operating system in conjunction with the 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 consists of two components, each physical component of the TOE must be considered when discussing the TOE cryptographic support. Both components of the TOE utilize cryptography to verify trusted updates, and the SMC uses cryptography to support its use of the TLS protocol to protect network communication.

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 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.

1.4.1.2.5 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.

1.4.1.2.6 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). Administrators can also access the TOE via a local console which provides limited functionality.

1.4.1.2.7 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's Linux-based operating system utilizes 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 a Security Administrator. The TOE also utilizes a dedicated, local network for communications between the TOE's components.

1.4.1.2.8 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.

1.4.1.2.9 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.

1.4.2 TOE Documentation

The following administrator and user guidance is available:

- Forcepoint Next Generation Firewall Common Criteria Evaluated Configuration Guide, 6.3.1 Rev E

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:
    - collaborative Protection Profile for Stateful Traffic Filter Firewalls, Version 1.0, 27 February 2015 (FWcPP10)
- Technical Decisions: TD090, TD0095, TD0111, TD0112, TD0113, TD0114, TD0116, TD0117, TD0126, TD0130, TD0143, TD0152, TD0153, TD0154, TD0156, TD0168, TD0169, TD0170, TD0181, TD0183, TD0184, TD0185, TD0187, TD0188, TD0199, TD0201, TD0226, TD0227, TD0228, TD0289, TD0290, TD0291

2.1 Conformance Rationale

The ST conforms to the FWcPP10. 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 FWcPP10 and this section reproduces only the corresponding Security Objectives for the operational environment for reader convenience. The FWcPP10 offers additional information about the identified security objectives, but that has not been reproduced here and the FWcPP10 should be consulted if there is interest in that material.

In general, the FWcPP10 has defined Security Objectives appropriate for firewalls and as such are applicable to the Forcepoint Next Generation Firewall TOE.

3.1 Security Objectives for the Operational Environment

- **OE.ADMIN_CREDENTIALS_SECURE** The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

- **OE.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 guidance documentation in a trusted manner.

- **OE.UPDATES** The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the FWcPP10. The FWcPP10 defines the following extended requirements and since they are not redefined in this ST, the FWcPP10 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- FAU_STG_EXT.1: Protected Audit Event Storage
- FCS_RBG_EXT.1: Random Bit Generation
- FCS_TLSC_EXT.2: TLS Client Protocol with authentication
- FCS_TLSS_EXT.1: TLS Server Protocol
- FFW_RUL_EXT.1: Stateful Traffic Filtering
- FFW_RUL_EXT.2: Stateful Filtering of Dynamic Protocols
- FIA_PMG_EXT.1: Password Management
- FIA_UAU_EXT.2: Password-based Authentication Mechanism
- FIA_UIA_EXT.1: User Identification and Authentication
- FIA_X509_EXT.1: X.509 Certificate Validation
- FIA_X509_EXT.2: X.509 Certificate Authentication
- FIA_X509_EXT.3: X.509 Certificate Requests
- FPT_APW_EXT.1: Protection of Administrator Passwords
- FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)
- FPT_TST_EXT.1: TSF testing
- FPT_TUD_EXT.1: 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 FWcPP10. The refinements and operations already performed in the FWcPP10 are not identified (e.g., highlighted) here, rather the requirements have been copied from the FWcPP10 and any residual operations have been completed herein. Of particular note, the FWcPP10 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 FWcPP10 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the FWcPP10 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The FWcPP10 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

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

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1: Audit Data Generation
	FAU_GEN.2: User identity association
	FAU_STG_EXT.1: Protected Audit Event Storage
FCS: Cryptographic support	FCS_CKM.1: Cryptographic Key Generation
	FCS_CKM.2: Cryptographic Key Establishment
	FCS_CKM.4: Cryptographic Key Destruction
	FCS_COP.1(1): Cryptographic Operation (AES Data Encryption/Decryption)
	FCS_COP.1(2): Cryptographic Operation (Signature Generation and Verification)
	FCS_COP.1(3): Cryptographic Operation (Hash Algorithm)
	FCS_COP.1(4): Cryptographic Operation (Keyed Hash Algorithm)
	FCS_RBG_EXT.1: Random Bit Generation
	FCS_TLSC_EXT.2: TLS Client Protocol with authentication
	FCS_TLSS_EXT.1: TLS Server Protocol
FDP: User data protection	FDP_RIP.2: Full Residual Information Protection
FFW: Firewall	FFW_RUL_EXT.1: Stateful Traffic Filtering
	FFW_RUL_EXT.2: Stateful Filtering of Dynamic Protocols
FIA: Identification and authentication	FIA_PMG_EXT.1: Password Management
	FIA_UAU.7: Protected Authentication Feedback
	FIA_UAU_EXT.2: Password-based Authentication Mechanism
	FIA_UIA_EXT.1: User Identification and Authentication
	FIA_X509_EXT.1: X.509 Certificate Validation
	FIA_X509_EXT.2: X.509 Certificate Authentication
	FIA_X509_EXT.3: X.509 Certificate Requests
FMT: Security management	FMT_MOF.1(1): Management of security functions behavior - Trusted Update
	FMT_MTD.1(1): Management of 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: Protection of Administrator Passwords
	FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)
	FPT_STM.1: Reliable Time Stamps

	FPT_TST_EXT.1: TSF testing
	FPT_TUD_EXT.1: Trusted update
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

Table 1 TOE Security Functional Components

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 comprising:

- Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).
- Security related configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- Resetting passwords (name of related user account shall be logged).
- Starting and stopping services (if applicable).
- [*no other actions*];

d) Specifically defined auditable events listed in **Table 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.2	None	None
FCS_CKM.4	None	None
FCS_COP.1(1)	None	None
FCS_COP.1(2)	None	None
FCS_COP.1(3)	None	None
FCS_COP.1(4)	None	None
FCS_RBG_EXT.1	None	None
FCS_TLSC_EXT.2	Failure to establish a TLS Session.	Reason for failure.
FCS_TLSS_EXT.1	Failure to establish a TLS Session.	Reason for failure.
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 Identifier of rule causing packet drop
FFW_RUL_EXT.2	FTP-related session establishment	Source and destination addresses. Source and destination ports.
FIA_PMG_EXT.1	None	None

FIA_UAU.7	None	None
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_X509_EXT.1	Unsuccessful attempt to validate a certificate.	Reason for failure.
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None	None
FMT_MOF.1(1)	Modification of the behavior of the audit functionality when Local Audit Storage Space is full.	
FMT_MTD.1(1)	Modification, deletion, generation/import of cryptographic keys.	
FMT_SMF.1	None	None
FMT_SMR.2	None	None
FPT_APW_EXT.1	None	None
FPT_SKP_EXT.1	None	None
FPT_STM.1	Changes to time.	The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FPT_TST_EXT.1	None	None
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	
FTA_SSL.4	The termination of an interactive session.	
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	
FTA_TAB.1	None	None
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	Identification of the claimed user identity.

Table 2 Audit events

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 2 Audit events**.

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 Protected Audit Event 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 according to FTP_ITC.1.

FAU_STG_EXT.1.2

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

FAU_STG_EXT.1.3

The TSF shall [***the Management Server will reject administrative actions that result in additional audit messages, the Log Server will send an alert to administrator and ultimately stop accepting new audit message from the Engine, the Engine will stop traffic and transition into an offline state until audit space is again available***] when the local storage space for audit data is full.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (FCS_CKM.1)

FCS_CKM.1.1

The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- ***RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3,***
- ***ECC schemes using 'NIST curves' [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4***]. (TD0291 applied)

5.1.2.2 Cryptographic Key Establishment (FCS_CKM.2)

FCS_CKM.2.1

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- ***RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, 'Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography',***
- ***Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography'***].

5.1.2.3 Cryptographic Key Destruction (FCS_CKM.4)

FCS_CKM.4.1

The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method

- For plaintext keys in volatile storage, the destruction shall be executed by a [***single overwrite consisting of [zeroes]***];
- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [***o logically addresses the storage location of the key and performs a [single] overwrite consisting of [a value that does not contain any CSP]***]

that meets the following: No Standard. (TD0130 applied)

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

FCS_COP.1(1).1

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

5.1.2.5 Cryptographic Operation (Signature Generation and Verification) (FCS_COP.1(2))

FCS_COP.1(2).1

The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [
- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits],
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits]]
that meet the following: [
- For RSA schemes: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSAPKCS1v1_5, ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3 (TD0116 applied),
- For ECDSA schemes: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 6 and Appendix D, Implementing 'NIST curves' [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4 (TD0199 applied)].

5.1.2.6 Cryptographic Operation (Hash Algorithm) (FCS_COP.1(3))

FCS_COP.1(3).1

The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384, SHA-512***] that meet the following: ISO/IEC 10118-3:2004.

5.1.2.7 Cryptographic Operation (Keyed Hash Algorithm) (FCS_COP.1(4))

FCS_COP.1(4).1

The TSF shall perform keyed-hash message authentication in accordance with a specified cryptographic algorithm [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384***] and cryptographic key sizes [***128, 160, 256, 384***] and message digest sizes [***160, 256, 384***] bits that meet the following: ISO/IEC 9797-2:2011, Section 7 'MAC Algorithm 2'.

5.1.2.8 Random Bit Generation (FCS_RBG_EXT.1)

FCS_RBG_EXT.1.1

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

FCS_RBG_EXT.1.2

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

5.1.2.9 TLS Client Protocol (FCS_TLSC_EXT.2)

FCS_TLSC_EXT.2.1

The TSF shall implement [***TLS 1.2 (RFC 5246)***] supporting the following ciphersuites (TD0226 applied):
[***-TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,***
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289].

FCS_TLSC_EXT.2.2

The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.2.3

The TSF shall only establish a trusted channel if the peer certificate is valid.

FCS_TLSC_EXT.2.4

The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [***secp256r1, secp384r1, secp521r1***] and no other curves.

FCS_TLSS_EXT.2.5

The TSF shall support mutual authentication using X.509v3 certificates.

5.1.2.10 TLS Server Protocol with mutual authentication (FCS_TLSS_EXT.1)

FCS_TLSS_EXT.1.1

The TSF shall implement [***TLS 1.2 (RFC 5246)***] supporting the following ciphersuites (TD0226 applied):
[***- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,***
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289].

FCS_TLSS_EXT.1.2

The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [***TLS 1.1,***]. (TD0156 applied)

FCS_TLSS_EXT.1.3

The TSF shall (TD0226 applied) [
- generate EC Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves;].

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 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 allow the definition of Stateful Traffic Filtering rules using the following network protocol fields:

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

- Source address
- Destination Address
- Transport Layer Protocol
- [***no other field***]

- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port
- and distinct interface.

FFW_RUL_EXT.1.3

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

FFW_RUL_EXT.1.4

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

FFW_RUL_EXT.1.5

The TSF shall:

a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [***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***].

FFW_RUL_EXT.1.6

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

a) The TSF shall drop and be capable of [***logging***] packets which are invalid fragments;

b) The TSF shall drop and be capable of [***logging***] fragmented packets which cannot be re-assembled completely;

c) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a broadcast network;

d) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a multicast network; The TSF shall drop and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;

e) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address 'reserved for future use' (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;

f) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as an 'unspecified address' or an address 'reserved for future definition and use' (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;

g) The TSF shall drop and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and

h) [***no other rules***].

FFW_RUL_EXT.1.7

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

a) The TSF shall drop and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received;

b) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is a link-local address;

c) The TSF shall drop and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received.

FFW_RUL_EXT.1.8

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

FFW_RUL_EXT.1.9

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

FFW_RUL_EXT.1.10

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

5.1.4.2 Stateful Filtering of Dynamic Protocols (FFW_RUL_EXT.2)

FFW_RUL_EXT.2.1

The TSF shall dynamically define rules or establish sessions allowing network traffic to flow for the following network protocols [***FTP***].

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:

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [***'!', '@', '#', '$', '%', '^', '&', '*', '(', ')'***] ;

b) Minimum password length shall be settable by the Security Administrator, and shall 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 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.

5.1.5.5 X.509 Certificate Validation (FIA_X509_EXT.1)

FIA_X509_EXT.1.1

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

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5 (TD0169 applied)***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

FIA_X509_EXT.1.2

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

5.1.5.6 X.509 Certificate Authentication (FIA_X509_EXT.2)

FIA_X509_EXT.2.1

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

FIA_X509_EXT.2.2

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

5.1.5.7 X.509 Certificate Requests (FIA_X509_EXT.3)

FIA_X509_EXT.3.1

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

FIA_X509_EXT.3.2

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

5.1.6 Security management (FMT)

5.1.6.1 Management of security functions behavior - Trusted Update (FMT_MOF.1(1))

FMT_MOF.1(1).1

The TSF shall restrict the ability to enable the functions perform manual update to Security Administrators.

5.1.6.2 Management of TSF Data (FMT_MTD.1(1))

FMT_MTD.1(1).1

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

5.1.6.3 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 configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [***digital signature***] capability prior to installing those updates (Per TD0090);
- [***o Ability to configure audit behavior,***
o Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1,
o Ability to configure the cryptographic functionality].

5.1.6.4 Restrictions on Security Roles (FMT_SMR.2)

FMT_SMR.2.1

The TSF shall maintain the roles: - Security Administrator.

FMT_SMR.2.2

The TSF shall be able to associate users with roles.

FMT_SMR.2.3

The TSF shall ensure that the conditions - The Security Administrator role shall be able to administer the TOE locally; - The Security Administrator role shall be able to administer the TOE remotely are satisfied.

5.1.7 Protection of the TSF (FPT)

5.1.7.1 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 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.

5.1.7.3 Reliable Time Stamps (FPT_STM.1)

FPT_STM.1.1

The TSF shall be able to provide reliable time stamps.

5.1.7.4 TSF testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: [***memory checks, KAT tests, and checksums of TOE binaries***].

5.1.7.5 Trusted update (FPT_TUD_EXT.1)

FPT_TUD_EXT.1.1

The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software and [***no other TOE firmware/software version***]. (TD0154 applied)

FPT_TUD_EXT.1.2

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

FPT_TUD_EXT.1.3

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

5.1.8 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

Refinement: 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

The TSF shall be capable of using [***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 [***transmitting audit records to an audit server***].

5.1.9.2 Trusted Path (FTP_TRP.1)

FTP_TRP.1.1

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

FTP_TRP.1.2

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.

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

Table 3 Assurance Components

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 categorisation 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
- 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 TOE's audit records include the required date, 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 2 Audit 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 clock provided by hardware and made available through the component's operating system to provide the timestamp for audit records. The SMC 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 Log Server immediately after generation of the record. The SMC Appliance Log Server stores Engine records and can also send those 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 can (if configured) send them to an external syslog server immediately after storing the records. When a connection to the external syslog server fails, the Management Server or Log Server will re-establish the connection and send NEW audit records to the syslog server.

When the NGFW Engine cannot transfer newly generated audit records to the Log Server (irrespective of the cause), the Engine will spool the records on disk. Once the NGFW Engine can again transfer audit records to the Log Server, it transfers the oldest records first and removes those from its spool, continuing until it has transferred all spooled records. If the NGFW Engine cannot transfer its audit records to the Log Server for an extended period of time, it may start to exhaust its available spool disk storage space (the size of this spool depends on the size of the disk in the NGFW Engine model, but the 2105/3305/6205 models provide 48/314/314GB of spool space, respectively). In such an event the NGFW Engine first sends alerts to notify the administrator that it has nearly exhausted its log spool, and once it exhausts its spool space, it follows the administrator defined behavior.

The Log Server, which aggregates audit records from NGFW Engines, writes incoming audit entries to the SMC Appliance disk storage (the Log Server's audit storage has its own 180GB logical partition in which to store the records). The proprietary protocol for synchronizing and managing the data between the NGFW Engine and the SMC Appliance Log Server starts with the Engine notifying the Log Server that there is a new log and then sending the new log entry to the Log Server. The Log Server stores the audit information as database files which are only accessible to a TOE administrator via the SMC Management Client. Again, only after the Log Server confirms successful receipt and storage of an audit entry does an Engine remove the audit entry from its spool.

The Log Server itself has a limited amount of disk storage in which to hold its database audit records. If the Log Server draws close to exhausting this space (specifically if it has fewer than 300MB of space remaining), it will alert the administrator (by creating an audit alert that the SMC Client displays to the administrator) warning of the low storage remaining (and the administrator can take action to remove old audit records). Should the Log Server continue to fill up its storage space and have less than 100MB of space remaining, it will stop accepting new audit messages from Engines.

The Management Server, like the Log Server, has a finite amount of space to store management audit records (10GB) on the SMC Appliance's disk. The Management Server stores its logs in a separate partition (the root partition), and should the Management Server begin to exhaust this space, it behaves similarly to the Log Server. When less than 300MB of space remains, the Management Server generates an administrative alert (displayed to the administrator through the SMC Client), and when less than 100MB of space remains, the Management Server will no longer allow the administrator to make changes (as such a change would result in an audit message that the Management Server no longer has sufficient space to safely store).

As mentioned above, the administrator defines the log spool policy for the NGFW Engines. This specifies the behavior of the NGFW Engines whenever its local log spool fills. 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 (if the Engine were part of a cluster). Once the spool situation has improved, the Engine/node returns automatically to an 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 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 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 **Table 2 Audit events**. The records include the required date, time, type, subject, outcome and event type values. The TOE also inserts into audit records all of the additional information described by **Table 2 Audit events**.
- 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 Engine are transmitted first to the SMC Appliance's Log Server and then (if configured by an administrator) forwarded to an external syslog server using TLS protected syslog.

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 cryptography from its 1.0.2n OpenSSL library (which incorporates the Forcepoint NGFW FIPS Object Module 2.0.14) for Engine integrity testing and Trusted Updates
- SMC Appliance
  - SMC Appliance uses cryptography from its 1.0.2n OpenSSL library (which incorporates the Forcepoint NGFW FIPS Object Module 2.0.13) for password hashing, integrity testing and verification of Trusted Updates
  - SMC Appliance's Management and Log Server use cryptography from the Forcepoint NGFW FIPS Java API 1.0.0 for the TOE TLS, CSR generation, and password hashing

SFR	Algorithm	NIST Standard	Cert#
FCS_COP.1(2) (Sign/Verify)	RSA Sign/Verify	FIPS 186-4, RSA	2777
	ECDSA Sign/Verify	FIPS 186-4, ECDSA	1339
FCS_COP.1(3) (Hash)	SHA Hashing	FIPS 180-4	4175
FCS_COP.1(4) (Keyed Hash)	HMAC-SHA	FIPS 198-1 & 180-4	3429

Table 6-1 NGFW Engine Forcepoint NGFW FIPS Object Module CAVP Certificates

SFR	Algorithm	NIST Standard	Cert#
FCS_COP.1(2) (Sign/Verify)	RSA Sign/Verify	FIPS 186-4, RSA	2776
	ECDSA Sign/Verify	FIPS 186-4, ECDSA	1338
FCS_COP.1(3) (Hash)	SHA Hashing	FIPS 180-4	4174
FCS_COP.1(4) (Keyed Hash)	HMAC-SHA	FIPS 198-1 & 180-4	3428

Table 6-2 SMC Appliance Forcepoint NGFW FIPS Object Module CAVP Certificates

SFR	Algorithm	NIST Standard	Cert#
FCS_CKM.1 (Key Gen)	RSA IFC Key Generation	FIPS 186-4, RSA	2775
	ECDSA ECC Key Generation	FIPS 186-4, ECDSA	1337
FCS_CKM.2 (Key Establishment)	RSA-based Key Exchange	Vendor affirm 800-56B SHS DRBG RSA	N/A
	ECC-based Key Exchange	SP 800-56A, KAS ECC (not a CVL) CVL 800-135 TLS	 158 1674
FCS_COP.1(1) (AES)	AES 128/256 CBC, GCM	FIPS 197, SP 800-38A/D	5166
FCS_COP.1(2) (Sign/Verify)	RSA Sign/Verify	FIPS 186-4, RSA	2775
	ECDSA Sign/Verify	FIPS 186-4, ECDSA	1337
FCS_COP.1(3) (Hash)	SHA Hashing	FIPS 180-4	4173
FCS_COP.1(4) (Keyed Hash)	HMAC-SHA	FIPS 198-1 & 180-4	3427
FCS_RBG_EXT.1 (Random)	DRBG Bit Generation	SP 800-90A	1944

Table 6-3 SMC Appliance Forcepoint NGFW FIPS Java API CAVP Certificates

6.2.1 NGFW Engine

The NGFW Engine uses the Forcepoint NGFW FIPS Object Module (see Table 6-1) for hashing, HMAC, and signature related operations.

6.2.2 SMC Appliance

The primary functions of the SMC Appliance are provided by the Management Server and the Log Server (described in following sections). However, the SMC Appliance performs the verification of trusted updates, independent of the operation of the Management Server and the Log Server. The operation of the TOE update feature is described in section 5.1, and the cryptography verifying the validity of the update comes from the SMC Appliance Forcepoint NGFW FIPS Object Module.

The SMC Appliance Forcepoint NGFW FIPS Object Module does not generate keys, but instead uses cryptography for (local console) password hashing, for power-up integrity testing, and for signature verification of TOE updates.

The SMC Appliance uses a software noise source, and uses the Linux kernel Random Number Generator (LKRNG) to provide output to user space (through /dev/random). The SMC Appliance uses /dev/random to instantiate its Forcepoint NGFW FIPS Java API's SHA-512 HASH_DRBG and generate keys. The Forcepoint NGFW FIPS Java API is used by the Management Server and Log Server as described below.

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 within TLS protocol session negotiations. Once the Management Server is initially installed, it creates a custom constructed ECDSA certificate for itself.

The Management Server utilizes the Forcepoint NGFW FIPS Java API for all encryption, decryption, hashing and signature operations associated with support for the TLS protocol. That Java library draws from /dev/random to instantiate is DRBG (a SHA-512 Hash_DRBG) as described in 6.2.2.

The Management Server also generates salts for password hashing and SHA-512 hashes user passwords (both when creating a new user or when verifying the password of an existing user).

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 Log 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 6-4.

The Management Server also accepts incoming administrative sessions (SMC Client connections) that are protected by TLS, and uses the cipher suites identified by **Table 6-5**.

6.2.2.2 Log Server

The SMC Appliance's Log Server communicates with the NGFW Engine over the dedicated network. Communication between the Log Server and the Management Server are initiated by the Management Server to transfer configuration data to the Log Server.

The Log Server utilizes the Forcepoint NGFW FIPS Java API library for all encryption, decryption, hashing and signature operations associated with support for the TLS protocol.

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

TLS_RSA_WITH_AES_128_CBC_SHA,
TLS_RSA_WITH_AES_256_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA,
TLS_RSA_WITH_AES_128_CBC_SHA256,
TLS_RSA_WITH_AES_256_CBC_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,
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

Table 6-4 Cipher suites to communicate with an External Syslog Server

The Forcepoint NGFW FIPS Java API used by the SMC Appliance (and thus used by both the Management and Log Servers) uses the same /dev/random to instantiate its Forcepoint NGFW FIPS Java API's SHA-512 HASH_DRBG and generate keys and that is described as part of the SMC Appliance in 6.2.2.

Similar to the Management Server, the Log Server also accepts incoming administrative sessions (SMC Client connections) that are protected by TLS. Again, the Log Server, like the Management Server, utilizes the below cipher suites to protect remote administrative sessions.

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

Table 6-5 Cipher suites to communicate with remote administrators

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 pseudo-random data
TLS pre-master secret	In Memory	Handshake done	Overwriting with zeros
TLS session key	In Memory	Close of session	Overwriting with zeros
Passwords	On Disk	Command	Overwriting once with pseudo-random data

Table 6-6 CSPs and Keys

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

- **FCS_CKM.1**: The SMC Appliance supports 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 or ECDH keys used during ECDHE_* and TLS_RSA_* TLS cipher suite negotiation.

Server Component	Client/Server/Both	DH key gen?	ECDH key gen?	RSA key gen?
Management Server	Both	N/A	Yes	Yes
Log Server	Both	N/A	Yes	Yes

  The TOE, when in the evaluated configuration, uses 256-bit encryption mode as the security strength. This setting causes the TOE to generate an ECDSA P-521 TLS server certificate, and also to use only AES-256 cipher suites for remote administrator connections. (Note that the TOE provides the administrator the flexibility to use AES-256 or AES-128 cipher suites for the TLS protected syslog export client. The TOE also provides the administrator the ability to generate or import either an ECDSA [P-256, P-384, or P-521] or RSA [2048, 3072, 4096] key to use for TLS Client or mutual authentication during TLS syslog export).

- **FCS_CKM.2**: See FCS_CKM.1

- **FCS_CKM.4**: The SMC components clear keys (TLS) from memory after those keys are no longer needed.

- **FCS_COP.1(1)**: The TOE 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 CAVP certificates are identified in the tables above in section 6.2 (**Table 6-1**, **Table 6-2**, and **Table 6-3**).

- **FCS_COP.1(2)**: The TOE supports the use of RSA with 2048 bit key sizes, and ECDSA with a key size of 256 bits or greater for cryptographic signatures (specifically NIST curves P-256, P-384, or P-521). The crypto modules providing the cryptographic signature services are identified in the tables above.
- **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 the tables above.
- **FCS_COP.1(4)**: The TOE supports keyed-hash message authentication using HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-384 using SHA-1/256/384 with 160/256/384-bit keys to produce a 160/256/384 output MAC. The SHA-1/256 and 384 algorithms have block sizes of 512 and 1024-bits respectively. The crypto modules providing the keyed-hash message authentication with the corresponding FIPS certificates are identified in the tables above. 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 (note that in TLS_ECDHE_* cipher suites, the TOE offers all three NIST curves and will select based upon what the peer specifies).
  - TOE integrity check (both the NGFW Engine and SMC Appliance check their file system integrity using HMAC-SHA-256 using a hardcoded 256-bit key), and
  - TLS 1.2 will use HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-384 with a 160/256/384 bit key, respectively.
- **FCS_RBG_EXT.1**: The TOE components perform random bit generation as described in section 6.2.2, SMC Appliance.
- **FCS_TLSC_EXT.2**: The TOE communicates with a remote audit server using the TLS protocol. The administrator can configure a reference identifier of DNS name, IP address, Common Name, Distinguished Name, SHA-1, SHA-256, SHA-512, and MD5 hash of the peer certificate. When configured with a DNS Name, the TOE will check the administrator configured value against the certificate's CN and SAN:DNS identifiers fields by first comparing the expected value against each SAN:DNS extension present in the certificate (if present), and if the TOE finds no SAN:DNS extensions, it will then compare the expected value against the certificate's CN. The TOE does not support certificate pinning. The administrator need not (and cannot) explicitly configure anything regarding the ECDHE curves, the TOE always presents P-256, P-384, and P-521 curves in its client hello. The TOE also supports handling of presented certificates that contain a wildcard in either the CN or in the SAN:DNS extension.
- **FCS_TLSS_EXT.1**: The TOE provides a TLS interface for GUI administration. The TOE's TLS server acts similarly to its TLS client in that the administrator need not (and cannot) explicitly configure anything regarding the ECDHE curves, the TOE will negotiate P-256, P-384, or P-521 curves based upon what the peer/client specifies it supports in its hello. Similarly the administrator need not and cannot specify the versions of TLS that the TOE's server will negotiate, the TOE only negotiates TLS v1.2 with clients.

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 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, any 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 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.

Protocol	Related RFC[1]	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

Table 6-7 Protocols & Fields Filtered by the TOE

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).

All received network packets are processed by the NGFW Engine software module before transmission. The NGFW software module does stateful filtering of the received network packets according to the configured traffic filtering rules. Protocol Agents are used for advanced processing of traffic that require special handling such as permitting an FTP data connection dynamically. The NGFW Engine software denies the traffic if the Protocol Agent cannot process the traffic. Incoming packets are dropped if a network packet cannot be processed due to insufficient memory. All incoming network packets are also discarded before the NGFW Engine software module has been loaded, and the NGFW Engine software module denies all traffic until the module has been configured. Network interfaces and routing are configured after the NGFW Engine software module has been loaded. If the configured firewall rules cannot be applied during startup, only the management network interface will be available and traffic through the firewall will be denied.

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 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, any 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 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 follows the standard TCP handshaking process (SYN, responding SYN-ACK, followed by ACK) to denote establishment of a stateful session, and the TOE's connection tracking will eliminate existing connections immediately, upon completion of the flow (in the case of TCP and FTP) or upon an inactivity timeout for the session.

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

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

Table 6-8 Connection Tracking Fields

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[2]. 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. An administrator can specify that a rule applies to a specific interface by specifying a zone, adding a given firewall interface to that zone, and then specifying that zone as either a source or destination address for the 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.

Rules relating to FFW_RULE_EXT.1.6

a) The NGFW Engine denies and allows logging packets which are invalid fragments;

b) The NGFW Engine denies and allows logging fragmented packets which cannot be re-assembled completely;

c) The NGFW Engine denies and allows logging packets where the source address of the network packet is defined as being on a broadcast network;

d) The NGFW Engine denies and allows logging packets where the source address of the network packet is defined as being on a multicast network;

The NGFW Engine denies and allows logging network packets where the source address of the network packet is defined as being a loopback address;

e) The NGFW Engine denies and allows logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address 'reserved for future use' (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;

f) The NGFW Engine denies and allows logging network packets where the source or destination address of the network packet is defined as an 'unspecified address' or an address 'reserved for future definition and use' (i.e. unicast addresses not in this address

[2] Additional target actions of “Continue” and “Jump” described later support complex security policies.

range: 2000::/3) as specified in RFC 3513 for IPv6;

g) The NGFW Engine denies and allows logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified;

Rules relating to FFW_RULE_EXT.1.7

a) The NGFW Engine denies packets where the source address is equal to the address of the network interface where the network packet was received;

b) The NGFW Engine denies packets where the source or destination address of the network packet is a linklocal address;

c) The NGFW Engine denies packets where the source address does not belong to the networks associated with the network interface where the network packet was received, as the Engine has an administrator defined set of networks associated with each configured network interface.

Table 6-9 Additional Stateful Filtering Rules

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 TOE can also track and maintain the number of half-open TCP connections, and the administrator can define a limit of the number of such connections (either for the Engine as a whole or for a specific rule). When the TOE detects that the threshold has been exceeded, the TOE denies additional SYN packets. The TOE will expire such half-open TCP connections after fifteen seconds by default, and the administrator can change this default by configuring the “TCP syn ack seen” timeout.

The 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. 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.
- **FFW_RUL_EXT.2**: The TOE performs stateful packet filtering on the FTP protocol.

6.5 Identification and authentication

The TOE authenticates local and remote 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 with 10 characters being the minimum in the evaluated configuration, 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.
- **FIA_X509_EXT.1**: OCSP and CRL is supported for X509v3 certificate validation. Certificates are validated as part of the authentication process when they are presented to the TOE and when they are loaded into the TOE. The following fields are verified as appropriate: signature, validity period, extended key usage, issuer's name, basic constraints (for CA certs).
- **FIA_X509_EXT.2**: Certificates are checked and if the TOE cannot establish a connection to determine the validity of the certificate, the TOE's behavior varies depending on revocation method. When handling a certificate bearing OCSP revocation but where the TOE cannot establish a connection with the OCSP responder, the TOE will not accept the certificate (and thus not establish the connection). When handling certificates bearing CRL information but where the TOE cannot establish a connection to the CRL Distribution Point location, the TOE will accept the certificate as valid. The TOE constructs the certificate path to a trusted certificate, and then verifies the signature, checks the revocation status, validity period, issuer's name, extended key usage and basic constraints for each certificate starting from the trusted certificate.
- **FIA_X509_EXT.3**: The TOE generates certificate requests and validates the CA used to sign the certificate.

6.6 Security management

The TOE provides an administrator role. User accounts that are associated with the administrator role are considered Security Administrators. Security Administrators can manage and configure audit configuration data, user and administrator security attributes (including [re]setting passwords, but not viewing an existing password), warning banner configuration, and cryptographic support settings. The TOE provides administrators the ability to configure session inactivity timeouts.

The SMC Appliance offers two administrative interfaces – command line and GUI (the NGFW Engine provides no administrator access at all). The SMC Appliance offers command line functions which are accessible via the CLI. The CLI is a text based interface which can be accessed from a directly connected terminal. These command line functions can be used to query the current SMC Appliance firmware version and update the SMC Appliance's firmware (an administrator must use the GUI to query and update NGFW Engine software), to manually set the SMC Appliance's time, and to configure the SMC CLI session time out.

The TOE also offers a non-CLI, remote interface for management. This remote interface offers access through the GUI client using TLS v1.2, and provides all management functionality except those commands available through the CLI.

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

- **FMT_MOF.1(1)**: Only the administrator can initiate product updates.

- **FMT_MTD.1(1)**: The TOE ensures that only security administrators can login to manage TSF data and configure TOE services. TSF data include audit data, cryptographic data, authentication data, configuration data, security attributes, session timeouts, and updates.
- **FMT_SMF.1**: Administrators can configure operations of the TOE through a GUI, including configuring cryptographic functionality, audit behavior, and services available prior to login. The local command line interface is used by administrators to verify and install TOE updates, manually set the time, and configure the CLI session timeout.
- **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 salted and hashed using SHA-512 when stored.

The communication between the NGFW Engine and the servers running on the SMC Appliance are all protected using a dedicated, local network connection.

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.

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 Engine receives its time updates from the SMC Management Server only. The SMC 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 memory checks (to check for bad or failing memory) and Known Answer Tests (KAT) for the cryptographic features provided by the NGFW Engine Forcepoint NGFW FIPS Object Module, SMC Appliance Forcepoint NGFW FIPS Object Module, and SMC Appliance Forcepoint NGFW FIPS Java API cryptographic libraries. These KAT tests cover operation of AES, RSA, ECDSA, DRBG, SHA 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 NGFW Engine will reboot as a result of the error, while the SMC Appliance will halt its boot as a result of a KAT error or any error in its verification of the HMAC-SHA-256 checksum of the SMC binaries upon system startup. And in a similar fashion, the NGFW Engine (using its Forcepoint NGFW FIPS Object Module implementation) uses a hardcoded key to verify the HMAC-SHA-256 checksum of the whole partition containing TOE binaries and if it finds an error, it halts its boot.

The TOE performs trusted updates for both of its components: the SMC management appliance and NGFW Engines. To update the TOE software of the NGFW Engine, an administrator can obtain an update from Forcepoint and then upload the update to the SMC. 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. Once the administrator selects to upgrade a specific NGFW Engine with a patch, the SMC will transfer that update to the NGFW Engine, the Engine will also verify the signature on the update (even though the SMC has already verified the update), the Engine will use that update package (which is an full filesystem 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, the Engine will reboot into that new partition.

To update the SMC itself, the administrator obtains an SMC patch from Forcepoint 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 on the SMC Appliance, and stored (synchronized) in both the Management Server's configuration database and in the Linux password database. Both locations store passwords in a non-plaintext form, and the TOE provides no interfaces to allow administrators to view plaintext passwords.
- **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 a 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, verifying certificate and certificate revocation validity, and measuring session timeouts. The TOE time can be set by administrator action or by 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 from Forcepoint or (in the case of NGFW Engine patches) by configuring the SMC Management Server to automatically download NGFW Engine 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 TOE 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 proposes the cipher suites listed in the **Table 6-4** when configured into its evaluated mode. The TOE supports the use of a client certificate (which an administrator can obtain from the internal CA, from an external CA, or can import), as the mechanism to authenticate the TOE to the syslog server. The administrator can also load trusted CA certificates to which the syslog server's certificate must chain.

The administrator's Client GUI is again a Java program providing a graphical user interface only. All decisions on whether the operation is allowed occur in the Management Server with which the Client GUI communicates. The Management Server only accepts TLS connections for the Client GUI and accepts the cipher suites listed in **Table 6-5** when communicating with the GUI.

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

- **FTP_ITC.1**: The TOE protects syslog communication from the Log 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.