# Forcepoint Network Security Platform 7.3 Security Target

Version 0.7
09/01/2026

*Prepared for:*

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

*Prepared By:*

Gossamer Laboratories
www.gossamersec.com

## LIST OF TABLES

# 1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is the Forcepoint Network Security Platform provided by Forcepoint, LLC. The TOE is being evaluated as a network device.

The Security Target contains the following additional sections:

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

***Conventions***

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In this ST, iteration may be 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. Alternately, a usually descriptive textual extension may be added after a slash (/) character to identify a specific iteration. For example, iterations of a requirement such as FCS_COP.1 might be identified as FCS_COP.1/HASH and FCS_COP.1/CRYPT.
    - 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 …"). An exception to this marking convention is the case of tables that have cells that content equivalent to nothing (e.g., "None", "No events specified", "NA", "No additional information") where the cell is simply left blank since that represents no meaningful change but is effectively a refinement. Another exception to this marking convention is where extraneous punctuation (e.g., extra brackets) may be omitted or incorrect punctuation (e.g., extra or missing periods) may be corrected, so long as the meaning is not changed.
- 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 Network Security Platform 7.3 Security Target

**ST Version** – Version 0.7

**ST Date** – 09/01/2026

## 1.2 TOE Reference

**TOE Identification** – Forcepoint Network Security Platform 7.3

**TOE Developer** – Forcepoint, LLC

**Evaluation Sponsor** – Forcepoint, LLC

## 1.3 TOE Overview

The Target of Evaluation (TOE) is Forcepoint Network Security Platform 7.3. The Forcepoint Network Security Platform (FNSP) is a stateful packet filtering firewall and VPN gateway. Being a stateful packet filtering firewall, the FNSP filters network traffic optimized through the use of stateful packet inspection. Being a VPN gateway, the FNSP provides IPsec VPN functionality to secure network data exchanged with peer gateways and VPN clients. The FNSP is intended to be used as a network perimeter security gateway that provides a controlled connection. The FNSP is centrally managed and generates audit records for security critical events.

## 1.4 TOE Description

The Forcepoint Network Security Platform is a stateful packet filtering firewall and VPN gateway. The Forcepoint Network Security Platform system is composed of the Security Engine (a physical or virtual appliance) and the Virtual Security Management Center (SMC). The Security Engine controls connectivity and information flow between internal and external connected networks. The Virtual SMC Appliance provides administrative functionality supporting the configuration and operation of Security Engines. Throughout the remainder of this document, references to the Security Engine are meant to reference the TOE's firewall engine, while references to the FNSP are meant to refer to the TOE as a whole.

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

The Security Engine provides VPN gateway capabilities, allowing the Engine to use IKE/IPsec to protect traffic exchanged with remote peer gateways (for a site-to-site VPN configuration) and with VPN clients.

The FNSP 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 Security Engine may be used in combination to provide a company with an overall network topology.

The Security Engine contains a hardened Linux operating system (with a 6.6 kernel) executing on a single or multi-processor Forcepoint hardware platform.

The Virtual SMC Appliance (or SMC) contains the Management Server and Log Server. Like the Security Engine, the SMC contains a hardened Linux-based operating system (which uses a 5.14 kernel) to support the management capabilities and allow for the operation and configuration of firewall engines.

### 1.4.1 TOE Architecture

The Forcepoint Network Security Platform system is a distributed TOE[1] consisting of the Security Management Center Appliance and one or more Security Engines under the control of the SMC. These Security Engines provide firewall functionality, VPN gateway functionality, and communicate securely with the SMC using its embedded cryptographic library for all cryptographic functionality. The Virtual SMC Appliance provides Management Server, Log Server functionality, and securely managed Engines. As the SMC utilizes both Java and C, the SMC relies

[1] The TOE is a distributed TOE consistent with Use Case 3 as defined in the NDcPP30e.

upon both Java and native cryptographic libraries for cryptographic functionality. In the evaluated configuration, the Virtual SMC Appliance communicates with Security Engines through a TLS-protected trusted channel.

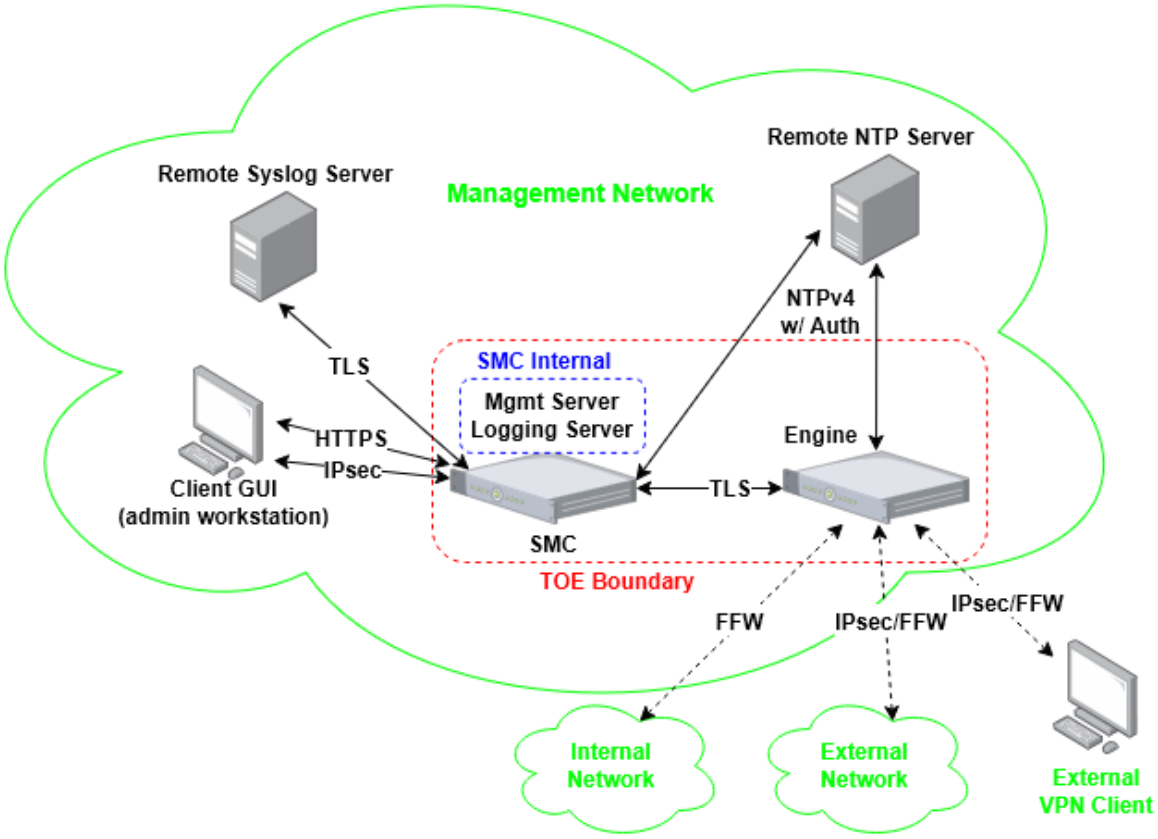


**Figure 1 TOE Components, Communication Paths and IT Environment.**

The Security Engines (a.k.a., the Engines) are responsible for performing all firewall packet handling, analysis and filtering that is provided by the Forcepoint Network Security Platform system, VPN gateway functionality (IKE/IPsec), as well as securely transmitting audit logs to the SMC's Log server.

The Management Server portion of the Virtual SMC Appliance provides the majority of the administrative capabilities in the Forcepoint Network Security Platform system through an Management Client GUI interface. The Virtual SMC Appliance provides a very limited console interface that allows administrators to verify and update the virtual SMC's software, to manually set the time, and configure the console timeout.

The Security Engines do not have local administrative interfaces, and can only be configured through the Virtual SMC Appliance. The Management Server is responsible for securely transferring the administrator defined configuration to Security Engines as the administrator makes configuration changes (these configuration changes are known as a 'security policy').

The Log Server in the Virtual SMC Appliance is responsible for securely collecting audit events from the Security 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 through a Management Client GUI (either the Forcepoint standalone Java Client installed from a Forcepoint provided installation package, through an HTTPS/TLS HTML5 web browser application, or either of those two methods but tunneled through IPsec). The Client GUI (along with the administrator's workstation on which the Client GUI runs), is part of the TOE's Operational Environment, and 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 Security Engine audit records that the Log Server has aggregated.

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 Virtual 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.
- **Management Server to External NTP Server communications** use SHA1 as the message digest algorithms for authentication with an NTP time source.
- **Security Engine to External NTP server communications** use SHA1 as the message digest algorithms for authentication with an NTP time source. Time on the Security Engine is updated by the SMC Management Server, or alternatively from an administrator configured NTP server.
- **Security Engine to Log Server communications** use the TLS-based trusted channel to protect the audit data transmitted from the Security Engine to the Log Server.
- **Security Engine to/from Management Server communications** use the TLS-based trusted channel to protect the configuration information exchanged between the Management Server and the Security 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 Security Engine. However, the Security Engine also polls for configuration changes on a regular basis.
- **Client GUI to Management and Log Server communications** uses HTTP/TLS or IPsec[2] to protect the communication over which remote administration actions occur.
- The **Security Engines** control connectivity and information flow between **internal and external connected networks** that they are protecting.
- The **Security Engines** encrypt information flow between the Engine and remote VPN peers (gateways and clients) with IKEv2/IPsec.

The cryptographic operations occurring as part of the communication on the Virtual SMC Appliance involving the Management Server and Log Server are performed using the Forcepoint Network Security Platform SMC FIPS Java API 2.0.1 (library). This provider provides the encryption, decryption, signing and hashing functions necessary to support the Virtual SMC Appliance use of the trusted channel mechanism and the trusted path mechanism. The Virtual SMC Appliance also uses the Forcepoint Network Security Platform SMC FIPS Library 3.0.7 (based on Oracle Linux 9 OpenSSL FIPS Provider 3.0.7-b27cdeb3ba51be46) to perform signature verification supporting the TOE trusted update mechanism. The Virtual SMC Appliance's NTP daemon uses cryptography from the Forcepoint Network Security Platform SMC FIPS Cryptographic Module for NTP 3.101.

The Security Engine utilizes its Forcepoint Network Security Platform Security Engine FIPS Library 3.1.2 (based upon OpenSSL 3.1.2 FIPS) to provide the encryption, decryption, signing and hashing functions necessary to support the Security Engine's trusted update mechanism and its TLS, ITT secure channel. The Security Engine also uses its Forcepoint Network Security Platform Security Engine FIPS Cryptographic Module 2.0 (based upon SafeZone FIPS Cryptographic Module 2.0) for IKE and uses its Forcepoint Network Security Platform Security Engine Cryptographic Kernel Module 3.0 for IPsec.

[2] Note: IPsec protection of Client GUI traffic requires a VPN client on the admin workstation and configuration of a Security Engine to terminate the tunnel

#### 1.4.1.1 Physical Boundaries

The TOE is composed of one or more Security Engine (physical or virtual) appliances and the Virtual SMC Appliance. Each of these have network connections to its environment, both to allow TLS protected management communications between the SMC and its engines, and network connections allowing the Security Engines to monitor and filter network traffic. The Virtual SMC Appliance provides all management functionality, while the Security Engines provide all firewall packet filtering.

The TOE is accessed and managed from the Forcepoint Security Management Center Client which can be used in a web browser or installed on a PC (admin workstation) in the environment, where the PC is expected to have a network pathway to the Virtual 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 Virtual SMC Appliance. The Security Engine does not send audit data directly to an external syslog server. Instead, a Security Engine passes all of its audit data to the Log Server on the Virtual SMC Appliance, which can (if configured) forward the data to the external syslog server.

The TOE's engines support IKE/IPsec connections with other engines as well as with external gateways.

An administrator can manually set the TOE's internal clock through the SMC console or via synchronization with an external NTP server. The Virtual SMC Appliance then configures the Security Engine's time to be in sync with itself. The Security Engine synchronizes only with the SMC, but can alternatively be configured separately to receive time from an NTP server directly.

The Security Engine utilizes its Forcepoint Network Security Platform Security Engine FIPS Library 3.1.2 (based upon OpenSSL 3.1.2 FIPS) to verify trusted engine software updates and uses its Forcepoint Network Security Platform Security Engine Cryptographic Kernel Module 3.0 to support IPsec connections. The Virtual SMC Appliance uses its Forcepoint Network Security Platform SMC FIPS Java API 2.0.1 to provide TLS (which protects the trusted channel mechanism and the trusted path mechanism) and uses its Forcepoint Network Security Platform SMC FIPS Library 3.0.7 (based on Oracle Linux 9 OpenSSL FIPS Provider 3.0.7-b27cdeb3ba51be46) 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 | Network I/O slots |
|---|---|---|---|---|---|
| 120 | Desktop<br>Intel® Atom C3338R (Goldmont) | 8 x GE RJ45 | 8 | 0 | 0 |
| 120W | Desktop<br>Intel® Atom C3338R (Goldmont) | 8 x GE RJ45 | 8 | 0 | 0 |
| 120WL | Desktop<br>Intel® Atom C3338R (Goldmont) | 8 x GE RJ45 | 8 | 0 | 0 |
| 120L | Desktop<br>Intel® Atom C3338R (Goldmont) | 8 x GE RJ45 | 8 | 0 | 0 |
| 125L | Desktop<br>Intel® Atom C3338R (Goldmont) | 8 x GE RJ45 | 8 | 0 | 0 |
| 60 | Desktop<br>Intel® Atom C3338R (Goldmont) | 4 x GE RJ45 | 4 | 0 | 0 |
| 60L | Desktop<br>Intel® Atom C3338R (Goldmont) | 4 x GE RJ45 | 4 | 0 | 0 |
| 61 | Desktop Intel® N50 (Alder Lake) | 4 x GE RJ45 | 4 | 0 | 0 |

| Model | Form factor/CPU | Fixed ports | 1G copper | 10G Fiber | Network I/O slots |
|---|---|---|---|---|---|
| 352 | Desktop Intel® Atom C5315 (Tremont) | 1x 2.5Gbps RJ45, 4x 10Gbps SFP+, 8 x GE RJ45 | 8 | 4 | 0 |
| 355 | Desktop Intel® Atom C5325 (Tremont) | 1x 2.5Gbps RJ45, 4x 10Gbps SFP+, 8 x GE RJ45 | 8 | 4 | 0 |
| 1202 | Desktop Intel® Atom P5322 (Tremont) | 2x 2.5Gbps RJ45, 8x 10Gbps SFP+, 8 x GE RJ45 | 8 | 8 | 0 |
| 2201 | 1U Intel® Xeon D-2123IT (Skylake) | 9x GE RJ45, 4x 10Gbps SFP+ | 9 to 17 | 4 to 12 | 1 |
| 2205 | 1U Intel® Xeon D-2145NT (Skylake) | 9x GE RJ45, 8x 10Gbps SFP+ | 9 to 17 | 8 to 16 | 1 |
| 2210 | 1U Intel® Xeon D-2177NT (Skylake) | 9x GE RJ45, 8x 10Gbps SFP+ | 9 to 16 | 8 to 16 | 1 |
| 2305 | 1U Intel® Xeon D-2876NT (Ice Lake) | 10x 1 GE RJ45, 4x 10 GE SFP+ | 1 to 17 | 0 to 16 | 2 |
| 2310 | 1U Intel® Xeon D-2896NT (Ice Lake) | 10x 1 GE RJ45, 4x 10 GE SFP+ | 1 to 17 | 0 to 16 | 2 |
| 3401 | 2U Intel® Xeon Silver 4210 (Cascade Lake) | 1x GE RJ45, 2x 10Gbps SFP+ | 1 to 65 | 2 to 66 | 8 |
| 3405 | 2U Intel® Xeon Silver 4216 (Cascade Lake) | 1x GE RJ45, 2x 10Gbps SFP+ | 1 to 65 | 2 to 66 | 8 |
| 3505 | 2U Intel® Xeon Silver 4416+ (Sapphire Rapids) | 1x GE RJ45, 2x 10Gbps SFP+ | 1 to 65 | 2 to 66 | 8 |
| 3510 | 2U Intel® Xeon Gold 6428N (Sapphire Rapids) | 1x GE RJ45, 2x 10Gbps SFP+ | 1 to 65 | 2 to 66 | 8 |
| ESXi 8.0 | Intel® Xeon Silver 4208 (Cascade Lake) | N/A | N/A | N/A | N/A |

The SMC model is as follows:

- Virtual SMC Appliance on ESXi 8.0 on Dell PowerEdge R440 with Intel Xeon® Silver 4208 (Cascade Lake)

The product was tested using the following configuration during the evaluation:

Desktop Firewall models

- 60 Desktop Atom C3338R (Goldmont)

1U Rack Mounted Firewall models

- 2210 1U Xeon D-2177NT (Skylake)

2U Rack Mounted Firewall models

- 3401 2U Xeon 4210 (Cascade Lake)

Virtual Security Engine Appliance

- VMWare ESXi 8.0 on Dell PowerEdge R440 with Intel Xeon® Silver 4208 (Cascade Lake)

#### 1.4.1.2 Logical Boundaries

This section summarizes the security functions provided by the Forcepoint Network Security Platform:

- Security audit
- Communication

- Cryptographic support
- User data protection
- Firewall
- Identification and authentication
- Security management
- Packet Filtering
- 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 Communication

The TOE is a distributed solution consisting of the Security Management Center and Security Engines. The Security Management Center can manage one or more Security Engines. The TOE uses a registration process to join Engines to an SMC.

#### 1.4.1.2.3 Cryptographic support

Because the TOE consists of distributed components, each physical component of the TOE must be considered when discussing the TOE cryptographic support. Both types of components (the SMC and its Engines) of the TOE utilize cryptography to verify trusted updates, for TLS protected management communications between the SMC and its Engines, for engine support of IPsec connections, and the SMC uses cryptography to support its use of the TLS protocol to protect administrative HTTPS connections and to protect network communications with external IT entities. Additionally, the TOE provides the ability to synchronize its time with a NTP server using NTPv4. The time data is protected by a SHA1 message digest.

#### 1.4.1.2.4 User data protection

The TOE ensures that all information flows from the TOE do not contain residual information from previous traffic. New packet data is used to overwrite any previous data in a buffer and any additional buffer space is padded with zeros before the packet is forwarded. Residual data is never transmitted from the TOE.

#### 1.4.1.2.5 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 Security Engine enforces the firewall security policy on all traffic that passes through the engine, via its internal or external network Ethernet interfaces.

#### 1.4.1.2.6 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.

The TOE supports X509v3 certificate validation during negotiation of TLS protected syslog and for secure communications between distributed TOE components (SMC and Security Engine). Certificates are validated as part of the authentication process when they are presented to the TOE and when they are loaded into the TOE.

#### 1.4.1.2.7 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 or in a web browser). Administrators can also access the TOE via a local console which provides limited functionality.

#### 1.4.1.2.8 Packet Filtering

Please see Section 1.4.1.2.5 Firewall for a description of the TOE's packet filtering mechanism.

#### 1.4.1.2.9 Protection of the TSF

The TOE provides a variety of means of protecting itself. The TOE protects communications between the SMC and engines using TLS as a trusted channel. Mutual authentication using client-side x.509v3 certificates is supported by the TLS communication between the distributed TOE components. The TOE performs self-tests that cover the correct operation of the TOE. It provides functions necessary to securely update the TOE. Its 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.

#### 1.4.1.2.10 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.

For VPN the TOE can terminate VPN client sessions that have been inactive for the administrator defined time interval. The TOE can deny VPN client sessions based upon location (as determined by IP address), time and day. The TOE can assign private IP addresses to VPN clients.

#### 1.4.1.2.11 Trusted path/channels

The TOE protects interactive communication with administrators using HTTPS/TLS for Client 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. Mutual authentication using client-side x.509v3 certificates is supported by the SMC TLS client for syslog over TLS.

The TOE protects data traffic sent between engines or to remote/external gateways and VPN clients using IPsec. The TOE can be configured to provide VPN Clients with access to the SMC's Management Server, allowing a remote administrator to securely access the SMC through a browser tunneled within IPsec.

The TOE protects communications between distributed components using a TLS-based trusted channel. The TOE uses a distinct TLS channel while registering new Engines with the SMC and once registered, the Engine and SMC communication is replaced with a different mutually-authenticated TLS channel to protect management communications.

### 1.4.2 TOE Documentation

The following administrator and user guidance is available:

- Forcepoint Network Security Platform Common Criteria Evaluated Configuration Guide, version 7.3.2, Revision A
- Forcepoint Network Security Platform Product Guide, version 7.3.0, Revision A
- Forcepoint Network Security Platform Installation Guide, version 7.3, Revision A

# 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 5, April 2017.
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
  - Part 3 Conformant
- Package Claims:
  - PP-Configuration for Network Devices, Stateful Traffic Filter Firewalls, and Virtual Private Network (VPN) Gateways, Version 2.0, 25 April 2024 consisting of the following components:
    - Base-PP: Protection Profile for Network Devices, Version 3.0e, 06 December 2023 (NDcPP30e)
    - PP-Module: PP-Module for Stateful Traffic Filter Firewalls, Version 1.4 + Errata 20200625, 25 June 2020 (STFFW14e)
    - PP-Module: PP-Module for VPN Gateways, Version 1.3, 16 August 2023 (VPNGW13)

| Package | Technical Decision | Applied | Notes |
|---|---|---|---|
| CPP_ND_V3.0E | TD0836 - NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1 | Yes | |
| CPP_ND_V3.0E | TD0868 - NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8 | Yes | |
| CPP_ND_V3.0E | TD0879 - NIT Decision: Correction of Chapter Headings in CPP_ND_V3.0E | Yes | |
| CPP_ND_V3.0E | TD0880 - NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1 | Yes | |
| CPP_ND_V3.0E | TD0886 - Clarification to FAU_STG_EXT.1 Test 6 | Yes | |
| CPP_ND_V3.0E | TD0899 - NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2 | Yes | |
| CPP_ND_V3.0E | TD0900 - NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3 | Yes | |
| CPP_ND_V3.0E | TD0921 - NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment | Yes | |
| CPP_ND_V3.0E | TD0923 - NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2 | Yes | |
| CPP_ND_V3.0E | TD0973 - NIT Technical Decision: FCS_(D)TLSS_EXT.1.3 Test 2 DHE Ciphersuite Conditionality | Yes | |
| CPP_ND_V3.0E | TD0990 - NIT Technical Decision: CTR_DRBG in FCS_RBG_EXT.1.2 | Yes | |
| CPP_ND_V3.0E | TD1033 - Sunset Dates for NDcPP Configurations | Yes | |
| CPP_ND_V3.0E | TD1052 - NIT Technical Decision: Separation of Test Definitions for DTLSv1.2 and v1.3 (Renegotiation) | No | DTLS is not claimed. |
| MOD_CPP_FW_V1.4E | TD0545 - NIT Technical Decision for Conflicting FW rules cannot be configured (extension of RfI#201837) | Yes | |
| MOD_CPP_FW_V1.4E | TD0551 - NIT Technical Decision for Incomplete Mappings | Yes | |

| Package | Technical Decision | Applied | Notes |
|---|---|---|---|
| | of OEs in FW Module v1.4+Errata | | |
| MOD_CPP_FW_V1.4E | TD0827 - Aligning MOD_CPP_FW_v1.4E with CPP_ND_V3.0E | Yes | |
| MOD_CPP_FW_V1.4E | TD0924 - NIT Technical Decision: FFW_RUL_EXT.1.2 Expected Rule Granularity Level | Yes | |
| MOD_VPNGW_V1.3 | TD0781 - Correction to FIA_PSK_EXT.3 EA for MOD_VPNGW_v1.3 | No | Password based PSK not supported |
| MOD_VPNGW_V1.3 | TD0811 - Correction to Referenced SFR in FIA_PSK_EXT.3 Test | No | Password based PSK not supported |
| MOD_VPNGW_V1.3 | TD0824 - Aligning MOD_VPNGW 1.3 with NDcPP 3.0E | Yes | |
| MOD_VPNGW_V1.3 | TD0941 - Updated Conformance Claims for MOD_VPNGW 1.3 | Yes | |
| MOD_VPNGW_V1.3 | TD0944 - Adding FIPS 186-5 in MOD_VPNGW_V1.3 | Yes | |
| MOD_VPNGW_v1.3 | TD0986 - Alignment of Updated IPsec Changes to MOD_VPNGW_V1.3 | Yes | |
| MOD_VPNGW_v1.3 | TD1034 - Clarification on RFC 8784 in FIA_PSK_EXT.1 | Yes | |

## 2.1 Conformance Rationale

The ST conforms to the NDcPP30e/STFFW14e/VPNGW13. 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 NDcPP30e/STFFW14e/VPNGW13 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDcPP30e/STFFW14e/VPNGW13 offers additional information about the identified security objectives, but that has not been reproduced here and the NDcPP30e/STFFW14e/VPNGW13 should be consulted if there is interest in that material.

In general, the NDcPP30e/STFFW14e/VPNGW13 has defined Security Objectives appropriate for a network device and as such are applicable to the Forcepoint Network Security Platform 7.3 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.COMPONENTS_RUNNING** (applies to distributed TOEs only)
For distributed TOEs, the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Security Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly.

**OE.CONNECTIONS** The TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.

**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. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.

**OE.NO_THRU_TRAFFIC_PROTECTION** The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.

**OE.PHYSICAL** Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

**OE.RESIDUAL_INFORMATION** The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.

**OE.TRUSTED_ADMIN** Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality.
For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.

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

**OE.VM_CONFIGURATION** (applies to vNDs only)
For vNDs, the Security Administrator ensures that the VS and VMs are configured to
- reduce the attack surface of VMs as much as possible while supporting ND functionality (e.g., remove unnecessary virtual hardware, turn off unused inter-VM communications mechanisms), and
- correctly implement ND functionality (e.g., ensure virtual networking is properly configured to support network traffic, management channels, and audit reporting).

The VS should be operated in a manner that reduces the likelihood that vND operations are adversely affected by virtualization features such as cloning, save/restore, suspend/resume, and live migration.
If possible, the VS should be configured to make use of features that leverage the VS's privileged position to provide additional security functionality. Such features could include malware detection through VM introspection, measured VM boot, or VM snapshot for forensic analysis.

# 4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDcPP30e/STFFW14e/VPNGW13. The NDcPP30e/STFFW14e/VPNGW13 defines the following extended requirements and since they are not redefined in this ST the NDcPP30e/STFFW14e/VPNGW13 should be consulted for more information in regard to those CC extensions.

**Extended SFRs:**

- NDcPP30e:FAU_GEN_EXT.1: Security Audit Data Generation
- NDcPP30e:FAU_STG_EXT.1: Protected Audit Event Storage
- NDcPP30e:FAU_STG_EXT.4: Protected Local Audit Event Storage for Distributed TOEs
- NDcPP30e:FAU_STG_EXT.5: Protected Remote Audit Event Storage for Distributed TOEs
- NDcPP30e:FCO_CPC_EXT.1: Component Registration Channel Definition
- NDcPP30e:FCS_HTTPS_EXT.1: HTTPS Protocol
- NDcPP30e:FCS_IPSEC_EXT.1: IPsec Protocol - per TD0868
- VPNGW13:FCS_IPSEC_EXT.1: IPsec Protocol - per TD0824
- NDcPP30e:FCS_NTP_EXT.1: NTP Protocol
- NDcPP30e:FCS_RBG_EXT.1: Random Bit Generation
- NDcPP30e:FCS_TLSC_EXT.1: TLS Client Protocol
- NDcPP30e:FCS_TLSC_EXT.2: TLS Client Support for Mutual Authentication
- NDcPP30e:FCS_TLSS_EXT.1: TLS Server Protocol
- NDcPP30e:FCS_TLSS_EXT.2: TLS Server Support for Mutual Authentication
- STFFW14e:FFW_RUL_EXT.1: Stateful Traffic Filtering
- STFFW14e:FFW_RUL_EXT.2: Stateful Filtering of Dynamic Protocols
- NDcPP30e:FIA_PMG_EXT.1: Password Management
- VPNGW13:FIA_PSK_EXT.1: Pre-Shared Key Composition
- VPNGW13:FIA_PSK_EXT.2: Generated Pre-Shared Keys
- NDcPP30e:FIA_UIA_EXT.1: User Identification and Authentication - per TD0900
- NDcPP30e:FIA_X509_EXT.1/ITT: X.509 Certificate Validation
- NDcPP30e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- VPNGW13:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- NDcPP30e:FIA_X509_EXT.2: X.509 Certificate Authentication
- VPNGW13:FIA_X509_EXT.2: X.509 Certificate Authentication
- NDcPP30e:FIA_X509_EXT.3: X.509 Certificate Requests
- VPNGW13:FIA_X509_EXT.3: X.509 Certificate Requests
- VPNGW13:FPF_RUL_EXT.1: Packet Filtering Rules
- NDcPP30e:FPT_APW_EXT.1: Protection of Administrator Passwords
- NDcPP30e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys)

- NDcPP30e:FPT_STM_EXT.1: Reliable Time Stamps
- NDcPP30e:FPT_TST_EXT.1: TSF testing - per TD0836
- VPNGW13:FPT_TST_EXT.1: TSF Testing - per TD0824
- VPNGW13:FPT_TST_EXT.3: Self-Test with Defined Methods
- NDcPP30e:FPT_TUD_EXT.1: Trusted update
- VPNGW13:FPT_TUD_EXT.1: Trusted Update - per TD0824
- NDcPP30e:FTA_SSL_EXT.1: TSF-initiated Session Locking
- VPNGW13:FTA_VCM_EXT.1: VPN Client Management

# 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 NDcPP30e/STFFW14e/VPNGW13. The refinements and operations already performed in the NDcPP30e/STFFW14e/VPNGW13 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDcPP30e/STFFW14e/VPNGW13 and any residual operations have been completed herein. Of particular note, the NDcPP30e/STFFW14e/VPNGW13 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 NDcPP30e/STFFW14e/VPNGW13. The NDcPP30e/STFFW14e/VPNGW13 should be consulted for the assurance activity definitions.

## 5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the Forcepoint Network Security Platform 7.3 TOE.

| Requirement Class | Requirement Component |
|---|---|
| **FAU: Security audit** | NDcPP30e:FAU_GEN.1: Audit Data Generation |
| | STFFW14e:FAU_GEN.1: Security Audit Data Generation |
| | VPNGW13:FAU_GEN.1/VPN: Audit Data Generation (VPN Gateway) |
| | NDcPP30e:FAU_GEN.2: User identity association |
| | NDcPP30e:FAU_GEN_EXT.1: Security Audit Data Generation |
| | NDcPP30e:FAU_STG_EXT.1: Protected Audit Event Storage |
| | NDcPP30e:FAU_STG_EXT.4: Protected Local Audit Event Storage for Distributed TOEs |
| | NDcPP30e:FAU_STG_EXT.5: Protected Remote Audit Event Storage for Distributed TOEs |
| **FCO: Communication** | NDcPP30e:FCO_CPC_EXT.1: Component Registration Channel Definition |
| **FCS: Cryptographic support** | NDcPP30e:FCS_CKM.1: Cryptographic Key Generation – per TD0921 |
| | VPNGW13:FCS_CKM.1/IKE: Cryptographic Key Generation (for IKE Peer Authentication) - per TD0944 |
| | NDcPP30e:FCS_CKM.2: Cryptographic Key Establishment |
| | NDcPP30e:FCS_CKM.4: Cryptographic Key Destruction |
| | NDcPP30e:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption) |
| | VPNGW13:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption) |
| | NDcPP30e:FCS_COP.1/Hash: Cryptographic Operation (Hash Algorithm) |
| | NDcPP30e:FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm) |
| | NDcPP30e:FCS_COP.1/SigGen: Cryptographic Operation (Signature Generation and Verification) – per TD0921 |
| | NDcPP30e:FCS_HTTPS_EXT.1: HTTPS Protocol |
| | NDcPP30e:FCS_IPSEC_EXT.1: IPsec Protocol - per TD0868 |
| | VPNGW13:FCS_IPSEC_EXT.1: IPsec Protocol - per TD0824 |
| | NDcPP30e:FCS_NTP_EXT.1: NTP Protocol |
| | NDcPP30e:FCS_RBG_EXT.1: Random Bit Generation |
| | NDcPP30e:FCS_TLSC_EXT.1: TLS Client Protocol |
| | NDcPP30e:FCS_TLSC_EXT.2: TLS Client Support for Mutual Authentication |
| | NDcPP30e:FCS_TLSS_EXT.1: TLS Server Protocol |
| | NDcPP30e:FCS_TLSS_EXT.2: TLS Server Support for Mutual Authentication |

| Requirement Class | Requirement Component |
|---|---|
| **FDP: User data protection** | STFFW14e:FDP_RIP.2: Full Residual Information Protection |
| **FFW: Firewall** | STFFW14e:FFW_RUL_EXT.1: Stateful Traffic Filtering |
| | STFFW14e:FFW_RUL_EXT.2: Stateful Filtering of Dynamic Protocols |
| **FIA: Identification and authentication** | NDcPP30e:FIA_AFL.1: Authentication Failure Management |
| | NDcPP30e:FIA_PMG_EXT.1: Password Management |
| | VPNGW13:FIA_PSK_EXT.1: Pre-Shared Key Composition |
| | VPNGW13:FIA_PSK_EXT.2: Generated Pre-Shared Keys |
| | NDcPP30e:FIA_UAU.7: Protected Authentication Feedback |
| | NDcPP30e:FIA_UIA_EXT.1: User Identification and Authentication - per TD0900 |
| | NDcPP30e:FIA_X509_EXT.1/ITT: X.509 Certificate Validation |
| | NDcPP30e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation |
| | VPNGW13:FIA_X509_EXT.1/Rev: X.509 Certificate Validation |
| | NDcPP30e:FIA_X509_EXT.2: X.509 Certificate Authentication |
| | VPNGW13:FIA_X509_EXT.2: X.509 Certificate Authentication |
| | NDcPP30e:FIA_X509_EXT.3: X.509 Certificate Requests |
| | VPNGW13:FIA_X509_EXT.3: X.509 Certificate Requests |
| **FMT: Security management** | NDcPP30e:FMT_MOF.1/Functions: Management of Security Functions Behaviour |
| | NDcPP30e:FMT_MOF.1/ManualUpdate: Management of security functions behaviour |
| | NDcPP30e:FMT_MTD.1/CoreData: Management of TSF Data |
| | NDcPP30e:FMT_MTD.1/CryptoKeys: Management of TSF Data |
| | VPNGW13:FMT_MTD.1/CryptoKeys: Management of TSF Data |
| | NDcPP30e:FMT_SMF.1: Specification of Management Functions - per TD0880 |
| | STFFW14e:FMT_SMF.1/FFW: Specification of Management Functions |
| | VPNGW13:FMT_SMF.1/VPN: Specification of Management Functions |
| | NDcPP30e:FMT_SMR.2: Restrictions on Security Roles |
| **FPF: Packet Filtering** | VPNGW13:FPF_RUL_EXT.1: Packet Filtering Rules |
| **FPT: Protection of the TSF** | NDcPP30e:FPT_APW_EXT.1: Protection of Administrator Passwords |
| | VPNGW13:FPT_FLS.1/SelfTest: Failure with Preservation of Secure State (Self-Test Failures) |
| | NDcPP30e:FPT_ITT.1: Basic internal TSF data transfer protection |
| | NDcPP30e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all symmetric keys) |
| | NDcPP30e:FPT_STM_EXT.1: Reliable Time Stamps |
| | NDcPP30e:FPT_TST_EXT.1: TSF testing - per TD0836 |
| | VPNGW13:FPT_TST_EXT.1: TSF Testing - per TD0824 |
| | VPNGW13:FPT_TST_EXT.3: Self-Test with Defined Methods |
| | NDcPP30e:FPT_TUD_EXT.1: Trusted update |
| | VPNGW13:FPT_TUD_EXT.1: Trusted Update - per TD0824 |
| **FTA: TOE access** | NDcPP30e:FTA_SSL.3: TSF-initiated Termination |
| | VPNGW13:FTA_SSL.3/VPN: TSF-Initiated Termination (VPN Headend) |
| | NDcPP30e:FTA_SSL.4: User-initiated Termination |
| | NDcPP30e:FTA_SSL_EXT.1: TSF-initiated Session Locking |
| | NDcPP30e:FTA_TAB.1: Default TOE Access Banners |
| | VPNGW13:FTA_TSE.1: TOE Session Establishment |
| | VPNGW13:FTA_VCM_EXT.1: VPN Client Management |
| **FTP: Trusted path/channels** | NDcPP30e:FTP_ITC.1: Inter-TSF trusted channel |
| | VPNGW13:FTP_ITC.1/VPN: Inter-TSF Trusted Channel (VPN Communications) |
| | NDcPP30e:FTP_TRP.1/Admin: Trusted Path |
| | NDcPP30e:FTP_TRP.1/Join: Trusted Path |

**Table 1 TOE Security Functional Components**

## 5.1.1 Security audit (FAU)

### 5.1.1.1 Audit Data Generation (NDcPP30e/STFFW14e:FAU_GEN.1)

**NDcPP30e/STFFW14e: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 Administrator account shall be logged if individual user accounts are required for administrators).
- Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- [***Resetting passwords (name of related Administrator account shall be logged)***];

d) Specifically defined auditable events listed in Table 2.

| Requirement | Auditable Events | Additional Content |
|---|---|---|
| **NDcPP30e:FAU_GEN.1** | None | None |
| **STFFW14e:FAU_GEN.1** | None | None |
| **VPNGW13:FAU_GEN.1/VPN** | No events specified. | N/A |
| **NDcPP30e:FAU_GEN.2** | None | None |
| **NDcPP30e:FAU_GEN_EXT.1** | None | None |
| **NDcPP30e:FAU_STG_EXT.1** | Configuration of local audit settings. | Identity of account making changes to the audit configuration. |
| **NDcPP30e:FAU_STG_EXT.4** | None | None |
| **NDcPP30e:FAU_STG_EXT.5** | None | None |
| **NDcPP30e:FCO_CPC_EXT.1** | Enabling communications between a pair of components. Disabling communications between a pair of components. | Identities of the endpoints pairs enabled or disabled. |
| **NDcPP30e:FCS_CKM.1** | None | None |
| **VPNGW13:FCS_CKM.1/IKE** | No events specified. | N/A |
| **VPNGW13:FCS_IPSEC_EXT.1** | Failure to establish an IPsec SA. | Reason for failure |
| **NDcPP30e:FCS_CKM.2** | None | None |
| **NDcPP30e:FCS_CKM.4** | None | None |
| **NDcPP30e:FCS_COP.1/DataEncryption** | None | None |
| **VPNGW13:FCS_COP.1/DataEncryption** | None | None |
| **NDcPP30e:FCS_COP.1/Hash** | None | None |
| **NDcPP30e:FCS_COP.1/KeyedHash** | None | None |
| **NDcPP30e:FCS_COP.1/SigGen** | None | None |
| **NDcPP30e:FCS_HTTPS_EXT.1** | Failure to establish a HTTPS Session. | Reason for failure. |
| **NDcPP30e:FCS_IPSEC_EXT.1** | Failure to establish an IPsec SA. | Reason for failure. |
| **NDcPP30e:FCS_NTP_EXT.1** | Configuration of a new time server Removal of configured time server | Identity if new/removed time server |
| **NDcPP30e:FCS_RBG_EXT.1** | None | None |
| **NDcPP30e:FCS_TLSC_EXT.1** | Failure to establish a TLS | Reason for failure. |

| Requirement | Auditable Events | Additional Content |
|---|---|---|
| | Session. | |
| **NDcPP30e:FCS_TLSC_EXT.2** | None | None |
| **NDcPP30e:FCS_TLSS_EXT.1** | Failure to establish a TLS Session. | Reason for failure. |
| **NDcPP30e:FCS_TLSS_EXT.2** | Failure to authenticate the client. | Reason for failure. |
| **STFFW14e:FDP_RIP.2** | None | None |
| **STFFW14e:FFW_RUL_EXT.1** | Application of rules configured with the 'log' operation | Source and destination addresses<br>Source and destination ports<br>Transport Layer Protocol<br>TOE Interface |
| **STFFW14e:FFW_RUL_EXT.2** | Dynamical definition of rule Establishment of a session | None |
| **NDcPP30e:FIA_AFL.1** | Unsuccessful login attempt limit is met or exceeded. | Origin of the attempt (e.g., IP address). |
| **NDcPP30e:FIA_PMG_EXT.1** | None | None |
| **NDcPP30e:FIA_UAU.7** | None | None |
| **NDcPP30e:FIA_UIA_EXT.1** | All use of identification and authentication mechanisms. | Origin of the attempt (e.g., IP address). |
| **NDcPP30e:FIA_X509_EXT.1/ITT** | Unsuccessful attempt to validate a certificate. Any addition, replacement or removal of trust anchors in the TOE's trust store | Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store |
| **NDcPP30e:FIA_X509_EXT.1/Rev** | Unsuccessful attempt to validate a certificate. Any addition, replacement or removal of trust anchors in the TOE's trust store | Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store |
| **VPNGW13:FIA_X509_EXT.1/Rev** | None | None |
| **NDcPP30e:FIA_X509_EXT.2** | None | None |
| **VPNGW13:FIA_X509_EXT.2** | None | None |
| **NDcPP30e:FIA_X509_EXT.3** | None | None |
| **VPNGW13:FIA_X509_EXT.3** | None | None |
| **NDcPP30e:FMT_MOF.1/Functions** | None | None |
| **NDcPP30e:FMT_MOF.1/ManualUpdate** | Any attempt to initiate a manual update. | None |
| **NDcPP30e:FMT_MTD.1/CoreData** | None | None |
| **NDcPP30e:FMT_MTD.1/CryptoKeys** | None | None |
| **VPNGW13:FMT_MTD.1/CryptoKeys** | None | None |
| **NDcPP30e:FMT_SMF.1** | All management activities of TSF data. | None |
| **STFFW14e:FMT_SMF.1/FFW** | All management activities of TSF data (including creation, modification and deletion of firewall rules). | None |
| **VPNGW13:FMT_SMF.1/VPN** | All administrative actions | No additional information. |
| **NDcPP30e:FMT_SMR.2** | None | None |
| **VPNGW13:FPF_RUL_EXT.1** | Application of rules configured with the 'log' operation | Source and destination addresses<br>Source and destination ports<br>Transport Layer Protocol |
| **NDcPP30e:FPT_APW_EXT.1** | None | None |
| **VPNGW13:FPT_FLS.1/SelfTest** | No events specified. | N/A |

| Requirement | Auditable Events | Additional Content |
|---|---|---|
| **NDcPP30e:FPT_ITT.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. |
| **NDcPP30e:FPT_SKP_EXT.1** | None | None |
| **NDcPP30e:FPT_STM_EXT.1** | Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1) | For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address). |
| **NDcPP30e:FPT_TST_EXT.1** | None | None |
| **VPNGW13:FPT_TST_EXT.1** | None | None |
| **VPNGW13:FPT_TST_EXT.3** | No events specified. | N/A |
| **NDcPP30e:FPT_TUD_EXT.1** | Initiation of update; result of the update attempt (success or failure). | None |
| **VPNGW13:FPT_TUD_EXT.1** | None | None |
| **NDcPP30e:FTA_SSL.3** | The termination of a remote session by the session locking mechanism. | None |
| **VPNGW13:FTA_SSL.3/VPN** | No events specified. | N/A |
| **NDcPP30e:FTA_SSL.4** | The termination of an interactive session. | None |
| **NDcPP30e:FTA_SSL_EXT.1** | (if 'terminate the session' is selected) The termination of a local session by the session locking mechanism. | None |
| **NDcPP30e:FTA_TAB.1** | None | None |
| **VPNGW13:FTA_TSE.1** | No events specified. | N/A |
| **VPNGW13:FTA_VCM_EXT.1** | No events specified. | N/A |
| **NDcPP30e:FTP_ITC.1** | Initiation of the trusted channel.<br>Termination of the trusted channel.<br>Failure of the trusted channel functions. | None<br>None<br>Reason for failure |
| **VPNGW13:FTP_ITC.1/VPN** | Initiation of the trusted channel.<br>Termination of the trusted channel.<br>Failure of the trusted channel functions. | No additional information.<br>No additional information.<br>Identification of the initiator and target of failed trusted channels establishment attempt |
| **NDcPP30e:FTP_TRP.1/Admin** | Initiation of the trusted path.<br>Termination of the trusted path.<br>Failure of the trusted path functions. | None<br>None<br>Reason for failure |
| **NDcPP30e:FTP_TRP.1/Join** | Initiation of the trusted path.<br>Termination of the trusted path.<br>Failure of the trusted path functions. | None |

**Table 2 Audit events**

**NDcPP30e/STFFW14e:FAU_GEN.1.2**

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, information specified in column three of Table 2.

#### 5.1.1.2 Audit Data Generation (VPN Gateway) (VPNGW13:FAU_GEN.1/VPN)

**VPNGW13:FAU_GEN.1.1/VPN**

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

a. Start-up and shutdown of the audit functions

b. Indication that TSF self-test was completed

c. Failure of self-test

d. All auditable events for the not specified level of audit; and

e. auditable events defined in **Table 2 Audit events**.

**VPNGW13:FAU_GEN.1.2/VPN**

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

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

b. For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, additional information defined in **Table 2 Audit events** for each auditable event, where applicable.

#### 5.1.1.3 User identity association (NDcPP30e:FAU_GEN.2)

**NDcPP30e: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.

#### 5.1.1.4 Security Audit Data Generation (NDcPP30e:FAU_GEN_EXT.1)

**NDcPP30e:FAU_GEN_EXT.1.1**

The TSF shall be able to generate audit records for each TOE component. The audit records generated by the TSF of each TOE component shall include the subset of security relevant audit events which can occur on the TOE component.

#### 5.1.1.5 Protected Audit Event Storage (NDcPP30e:FAU_STG_EXT.1)

**NDcPP30e: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.

**NDcPP30e:FAU_STG_EXT.1.2**

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

- ***The TOE shall be a distributed TOE that stores audit data on the following TOE components: [Virtual SMC Appliance],***
- ***The TOE shall be a distributed TOE with storage of audit data provided externally for the following TOE components: [the Security Engine transfers audit records to the Virtual SMC Appliance]***

].

**NDcPP30e:FAU_STG_EXT.1.3**

The TSF shall maintain a [***log file, [log files in a spool space]***] of audit records in the event that an interruption of communication with the remote audit server occurs.

**NDcPP30e:FAU_STG_EXT.1.4**

The TSF shall be able to store [***persistent***] audit records locally with a minimum storage size of [***180GB (for the SMC's Log Server), and 10GB (for the SMC's Management Server)***].

**NDcPP30e:FAU_STG_EXT.1.5**

The TSF shall [

- ***[the SMC Management Server will reject configuration changes that result in additional audit messages until action is taken by the administrator to delete local audit data,***
- ***the SMC Log Server will send an alert to the administrator and ultimately stop accepting new audit messages from the Security Engine]***

] when the local storage space for audit data is full.

**NDcPP30e:FAU_STG_EXT.1.6**

The TSF shall provide the following mechanisms for administrative access to locally stored audit records [***ability to view locally***].

#### 5.1.1.6 Protected Local Audit Event Storage for Distributed TOEs (NDcPP30e:FAU_STG_EXT.4)

**NDcPP30e:FAU_STG_EXT.4.1**

The TSF of each TOE component which stores security audit data locally shall perform the following actions when the local storage space for audit data is full: **[**

- ***the SMC Management Server [[will reject configuration changes that result in additional audit messages until action is taken by the administrator to delete local audit data]],***
- ***the SMC Log Server [[will send an alert to the administrator and ultimately stop accepting new audit messages from the Security Engine]],***
- ***the Security Engine [[will stop traffic and transition into an offline state until audit space is again available]]***

**].**

#### 5.1.1.7 Protected Remote Audit Event Storage for Distributed TOEs (NDcPP30e:FAU_STG_EXT.5)

**NDcPP30e:FAU_STG_EXT.5.1**

Each TOE component which does not store security audit data locally shall be able to buffer security audit data locally until it has been transferred to another TOE component that stores or forwards it. All transfer of audit records between TOE components shall use a protected channel according to [***FPT_ITT.1***].

### 5.1.2 Communication (FCO)

#### 5.1.2.1 Component Registration Channel Definition (NDcPP30e:FCO_CPC_EXT.1)

**NDcPP30e:FCO_CPC_EXT.1.1**

The TSF shall require a Security Administrator to enable communications between any pair of TOE components before such communication can take place.

**NDcPP30e:FCO_CPC_EXT.1.2**

The TSF shall implement a registration process in which components establish and use a communications channel that uses [

- ***A channel that meets the secure registration channel requirements in FTP_TRP.1/Join***

] for at least TSF data.

**NDcPP30e:FCO_CPC_EXT.1.3**

The TSF shall enable a Security Administrator to disable communications between any pair of TOE components.

### 5.1.3 Cryptographic support (FCS)

#### 5.1.3.1 Cryptographic Key Generation – per TD0921 (NDcPP30e:FCS_CKM.1)

**NDcPP30e: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, 3072, 4096] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;***
- ***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, or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6;***
- ***FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.1;***
- ***FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526]***

]. (TD0921 applied)

#### 5.1.3.2 Cryptographic Key Generation (for IKE Peer Authentication) - per TD0944 (VPNGW13:FCS_CKM.1/IKE)

**VPNGW13:FCS_CKM.1.1/IKE**

The TSF shall generate asymmetric cryptographic keys used for IKE peer authentication in accordance with a specified cryptographic key generation algorithm: [

- ***FIPS PUB 186-5, 'Digital Signature Standard (DSS)', Appendix A.1 for RSA schemes,***
- ***FIPS PUB 186-5, 'Digital Signature Standard (DSS)', Appendix A.2 for ECDSA schemes and implementing 'NIST curves' P-384 and [P-256, P-521]***]

and [

- ***FFC Schemes using 'safe-prime' groups that meet the following: 'NIST Special Publication 800-56A Revision 3, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' and [RFC 3526]***

] and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits. (TD0944 applied)

#### 5.1.3.3 Cryptographic Key Establishment (NDcPP30e:FCS_CKM.2)

**NDcPP30e:FCS_CKM.2.1**

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

- ***Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography';***
- ***FFC Schemes using "FIPS 186-Type" parameter-size sets that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";***
- ***FFC Schemes using 'safe-prime' groups that meet the following: 'NIST Special Publication 800-56A Revision 3, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' and [groups listed in RFC 3526]***

].

#### 5.1.3.4 Cryptographic Key Destruction (NDcPP30e:FCS_CKM.4)

**NDcPP30e: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], destruction of reference to the key directly followed by a request for garbage collection***];

- For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [
  - ***logically addresses the storage location of the key and performs a [single] overwrite consisting of [[pseudo-random data sourced from /dev/random]]***]

that meets the following: No Standard.

#### 5.1.3.5 Cryptographic Operation (AES Data Encryption/Decryption) (NDcPP30e:FCS_COP.1/DataEncryption)

**NDcPP30e:FCS_COP.1.1/DataEncryption**

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, 192 bits***] that meet the following: AES as specified in ISO 18033-3, [***CBC as specified in ISO 10116, GCM as specified in ISO 19772***].

#### 5.1.3.6 Cryptographic Operation (AES Data Encryption/Decryption) (VPNGW13:FCS_COP.1/DataEncryption)

**VPNGW13:FCS_COP.1.1/DataEncryption**

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC, GCM***] and [***no other***] mode and cryptographic key sizes [***128 bits, 256 bits***] and [***192 bits***] that meet the following: AES as specified in ISO 18033-3, [***CBC as specified in ISO 10116, GCM as specified in ISO 19772***], and [***no other standards***].

#### 5.1.3.7 Cryptographic Operation (Hash Algorithm) (NDcPP30e:FCS_COP.1/Hash)

**NDcPP30e:FCS_COP.1.1/Hash**

The TSF shall perform cryptographic hashing services in accordance with a specified cryptographic algorithm [***SHA-1, SHA-256, SHA-384, SHA-512***] and message digest sizes [***160, 256, 384, 512***] bits that meet the following: ISO/IEC 10118-3:2004.

#### 5.1.3.8 Cryptographic Operation (Keyed Hash Algorithm) (NDcPP30e:FCS_COP.1/KeyedHash)

**NDcPP30e:FCS_COP.1.1/KeyedHash**

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

#### 5.1.3.9 Cryptographic Operation (Signature Generation and Verification) (NDcPP30e:FCS_COP.1/SigGen)

**NDcPP30e:FCS_COP.1.1/SigGen**

The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [
- ***RSA Digital Signature Algorithm,***
- ***Elliptic Curve Digital Signature Algorithm***

] and cryptographic key sizes [
- ***For RSA: [modulus 2048, 3072, or 4096],***
- ***For ECDSA: [256, 384, or 521 bits]***

] that meet the following: [
- ***For RSA schemes: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 5.5, using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,***
- ***For ECDSA schemes implementing [P-256, P-384, P-521] curves that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 6 and Appendix D,***

***Implementing 'NIST Recommended' curves; or FIPS PUB 186-5, 'Digital Signature Standard (DSS)', Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3, 'IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms', Section 6.6***

].(TD0921 applied)

#### 5.1.3.10 HTTPS Protocol (NDcPP30e:FCS_HTTPS_EXT.1)

**NDcPP30e:FCS_HTTPS_EXT.1.1**

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

**NDcPP30e:FCS_HTTPS_EXT.1.2**

The TSF shall implement HTTPS using TLS.

#### 5.1.3.11 IPsec Protocol - per TD0868 (NDcPP30e:FCS_IPSEC_EXT.1)

**NDcPP30e:FCS_IPSEC_EXT.1.1**

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

**NDcPP30e:FCS_IPSEC_EXT.1.2**

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

**NDcPP30e:FCS_IPSEC_EXT.1.3**

The TSF shall implement [***tunnel mode***].

**NDcPP30e:FCS_IPSEC_EXT.1.4**

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [***AES-CBC-128 (RFC 3602), AES-CBC-192 (RFC 3602), AES-CBC-256 (RFC 3602), AES-GCM-128 (RFC 4106), AES-GCM-192 (RFC 4106), AES-GCM-256 (RFC 4106)***] together with a Secure Hash Algorithm (SHA)-based HMAC [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512***].

**NDcPP30e:FCS_IPSEC_EXT.1.5**

The TSF shall implement the protocol: [

- ***IKEv2 as defined in RFC 7296 [with mandatory support for NAT traversal as specified in RFC 7296, section 2.23], and [RFC 4868 for hash functions]***

].

**NDcPP30e:FCS_IPSEC_EXT.1.6**

The TSF shall ensure the encrypted payload in the [***IKEv2***] protocol uses the cryptographic algorithms [***AES-CBC-128, AES-CBC-192, AES-CBC-256 (specified in RFC 3602), AES-GCM-256 (specified in RFC 5282)***].

**NDcPP30e:FCS_IPSEC_EXT.1.7**

The TSF shall ensure that [

- ***IKEv2 SA lifetimes can be configured by a Security Administrator based on [***
  - ***length of time, where the time values can be configured between [1 minute] and [24 hours]]***

]. (TD0868 applied)

**NDcPP30e:FCS_IPSEC_EXT.1.8**

The TSF shall ensure that [

- ***IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [***
  - ***number of bytes;***
  - ***length of time, where the time values can be configured between [1 minute] and [8 hours]]***

]. (TD0868 applied)

**NDcPP30e:FCS_IPSEC_EXT.1.9**

The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ('x' in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [**224, 256, 300, 384, 512**] bits.

**NDcPP30e:FCS_IPSEC_EXT.1.10**

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

- ***at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash***

].

**NDcPP30e:FCS_IPSEC_EXT.1.11**

The TSF shall ensure that IKE protocols implement DH Group(s) [

- ***[14 (2048-bit MODP), 15 (3072-bit MODP), 16 (4096-bit MODP), 17 (6144-bit MODP), 18 (8192-bit MODP)] according to RFC 3526,***
- ***[19 (256-bit Random ECP), 20 (384-bit Random ECP), 21 (521-bit Random ECP)] according to RFC 5114***

].

**NDcPP30e:FCS_IPSEC_EXT.1.12**

The TSF shall be able to ensure that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [***IKEv2 IKE_SA***] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [***IKEv2 CHILD_SA***] connection.

**NDcPP30e:FCS_IPSEC_EXT.1.13**

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

**NDcPP30e:FCS_IPSEC_EXT.1.14**

The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: [***SAN: IP address, SAN: Fully Qualified Domain Name (FQDN), SAN: user FQDN, Distinguished Name (DN)***] and [***no other reference identifier type***].

### 5.1.3.12 IPsec Protocol - per TD0824 & TD0986 (VPNGW13:FCS_IPSEC_EXT.1)

**VPNGW13:FCS_IPSEC_EXT.1.1**

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

**VPNGW13:FCS_IPSEC_EXT.1.2**

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

**VPNGW13:FCS_IPSEC_EXT.1.3**

The TSF shall implement [***tunnel mode***].

**VPNGW13:FCS_IPSEC_EXT.1.4**

The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [***AES-CBC-128, AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES-GCM-256 (specified in RFC 4106)***] and [***AES-CBC-192 (specified in RFC 3602), AES-GCM-192 (specified in RFC 4106)***] together with a Secure Hash Algorithm (SHA)-based HMAC [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512***].

**VPNGW13:FCS_IPSEC_EXT.1.5**

The TSF shall implement the protocol: [

- ***IKEv2 as defined in RFC 7296 and: [with mandatory support for NAT traversal as specified in RFC 7296, section 2.23] and [RFC 4868 for hash functions]***

]. (TD0824 applied)

**VPNGW13:FCS_IPSEC_EXT.1.6**

The TSF shall ensure the encrypted payload in the [***IKEv2***] protocol uses the cryptographic algorithms [***AES-CBC-128, AES-CBC-192, AES-CBC-256 (specified in RFC 3602), AES-GCM-256 (specified in RFC 5282)***].

**VPNGW13:FCS_IPSEC_EXT.1.7**

The TSF shall ensure that [

- ***IKEv2 SA lifetimes can be configured by a Security Administrator based on [***
  - ***length of time, where the time values can be configured between [1 minute] and [24 hours]]***

]. (per TD0986)

**VPNGW13:FCS_IPSEC_EXT.1.8**

The TSF shall ensure that [

- ***IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [***
  - ***number of bytes;***
  - ***length of time, where the time values can be configured between [1 minute] and [8 hours]]***

]. (per TD0986)

**VPNGW13:FCS_IPSEC_EXT.1.9**

The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ('x' in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [**224, 256, 300, 340, 384**] bits.

**VPNGW13:FCS_IPSEC_EXT.1.10**

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

- ***at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash***

].

**VPNGW13:FCS_IPSEC_EXT.1.11**

The TSF shall ensure that IKE protocols implement DH Groups

- 20 (384-bit Random ECP) according to RFC 5114 and
- [***[14 (2048-bit MODP), 15 (3072-bit MODP), 16 (4096-bit MODP), 17 (6144-bit MODP), 18 (8192-bit MODP)] according to RFC 3526,***
- ***[19 (256-bit Random ECP), 21 (521-bit Random ECP)] according to RFC 5114***].

(per TD0986)

**VPNGW13:FCS_IPSEC_EXT.1.12**

The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [***IKEv2 IKE_SA***] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [***IKEv2 CHILD_SA***] connection.

**VPNGW13:FCS_IPSEC_EXT.1.13**

The TSF shall ensure that [***IKEv2***] protocols perform peer authentication using [***RSA, ECDSA***] that use X.509v3 certificates that conform to RFC 4945 and [***Pre-shared Keys that conform to RFC 8784***]. (TD0824 applied)

**VPNGW13:FCS_IPSEC_EXT.1.14**

The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: Distinguished Name (DN), [***SAN: IP address, SAN: Fully Qualified Domain Name (FQDN), SAN: user FQDN***].

### 5.1.3.13 NTP Protocol (NDcPP30e:FCS_NTP_EXT.1)

**NDcPP30e:FCS_NTP_EXT.1.1**

The TSF shall use only the following NTP version(s) [***NTP v4 (RFC 5905)***].

**NDcPP30e:FCS_NTP_EXT.1.2**

The TSF shall update its system time using [

- ***Authentication using [SHA1] as the message digest algorithm(s);***

].

**NDcPP30e:FCS_NTP_EXT.1.3**

The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

**NDcPP30e:FCS_NTP_EXT.1.4**

The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

#### 5.1.3.14 Random Bit Generation – per TD0990 (NDcPP30e:FCS_RBG_EXT.1)

**NDcPP30e: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 [SHA-512], CTR_DRBG (AES)***].

**NDcPP30e: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, [one] platform-based noise source***] with a minimum of [***256 bits, 384 bits***] of entropy at least equal to [
- ***the greatest security strength, according to ISO/IEC 18031:2011Table C.1 'Security Strength Table for Hash Functions'***,
- ***the seed length according to ISO/IEC 18031:2011 Table C.2 — "Security strengths, Entropy and Seed length requirements for the AES-128, 192 and 256 Block Cipher"***

]. (TD0990 applied)

#### 5.1.3.15 TLS Client Protocol (NDcPP30e:FCS_TLSC_EXT.1)

**NDcPP30e:FCS_TLSC_EXT.1.1**
The TSF shall implement [***TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:[
**Syslog over TLS (SMC as Client)**
***TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,***
***TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,***
***TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,***
***TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288,***
***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,***
***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,***
***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289***

**FPT_ITT (Engine as Client)**
***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***

**FPT_ITT (SMC as Client)**
***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***
] and no other ciphersuites.

**NDcPP30e:FCS_TLSC_EXT.1.2**
The TSF shall verify that the presented identifier matches [***the reference identifier per RFC 6125 Section 6, IPv4 address in the SAN***].

**NDcPP30e:FCS_TLSC_EXT.1.3**
The TSF shall not establish a trusted channel if the server certificate is invalid: [***without any administrator override mechanism.***].

**NDcPP30e:FCS_TLSC_EXT.1.4**
The TSF shall [***present the Supported Groups Extension with the following curves/groups: [secp256r1, secp384r1, secp521r1] and no other curves/groups***] in the Client Hello.

**NDcPP30e:FCS_TLSC_EXT.1.5**
The TSF shall [***present the signature_algorithms extension with support for the following algorithms:***
**Syslog over TLS (SMC as Client)**
**rsa_pkcs1 with sha256(0x0401),**
**rsa_pkcs1 with sha384(0x0501),**
**rsa_pkcs1 with sha512(0x0601),**
**ecdsa_secp256r1 with sha256(0x0403),**
**ecdsa_secp384r1 with sha384(0x0503),**

**ecdsa_secp521r1 with sha512(0x0603),**
**rsa_pss_rsae with sha256(0x0804),**
**rsa_pss_rsae with sha384(0x0805),**
**rsa_pss_rsae with sha512(0x0806),**
**rsa_pss_pss with sha256(0x0809),**
**rsa_pss_pss with sha384(0x080a),**
**rsa_pss_pss with sha512(0x080b,**

**FPT_ITT (Engine as Client)**
**ecdsa_secp384r1 with sha384(0x0503)**

**FPT_ITT (SMC as Client)**
**ecdsa_secp384r1 with sha384(0x0503)**
***] and no other algorithms***]

**NDcPP30e:FCS_TLSC_EXT.1.6**
The TSF [***provides***] the ability to configure the list of supported ciphersuites as defined in NDcPP30e:FCS_TLSC_EXT.1.1.

**NDcPP30e:FCS_TLSC_EXT.1.7**
The TSF shall prohibit the use of the following extensions:
- Early data extension
- Post-handshake client authentication according to RFC 8446, Section 4.2.6.

**NDcPP30e:FCS_TLSC_EXT.1.8**
The TSF shall [***not use PSKs***].

**NDcPP30e:FCS_TLSC_EXT.1.9**
The TSF shall [***support TLS 1.2 secure renegotiation through use of the 'renegotiation_info' TLS extension in accordance with RFC 5746***].

#### 5.1.3.16 TLS Client Support for Mutual Authentication (NDcPP30e:FCS_TLSC_EXT.2)

**NDcPP30e:FCS_TLSC_EXT.2.1**
The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

#### 5.1.3.17 TLS Server Protocol (NDcPP30e:FCS_TLSS_EXT.1)

**NDcPP30e:FCS_TLSS_EXT.1.1**
The TSF shall implement [***TLS 1.2 (RFC 5246)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

**SMC GUI**
***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***

**FPT_ITT (Engine as Server)**
***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***

**FPT_ITT (SMC as Server)**
***TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***
]
and no other ciphersuites.

**NDcPP30e:FCS_TLSS_EXT.1.2**
The TSF shall authenticate itself using X.509 certificate(s) using [***ECDSA over NIST curves [secp256r1, secp384r1, secp521r1(WebUI), secp384r1(Engine as Server, SMC as Server)] and no other curves***].

**NDcPP30e:FCS_TLSS_EXT.1.3**
The TSF shall perform key exchange using: [
- ***EC Diffie-Hellman key agreement over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves***

].

**NDcPP30e:FCS_TLSS_EXT.1.4**

The TSF shall support [***no session resumption(Engine as a Server), session resumption based on session IDs according to RFC 5246 (TLS1.2)(SMC as a Server, SMC GUI)***].

**NDcPP30e:FCS_TLSS_EXT.1.5**

The TSF [***provides***] the ability to configure the list of supported ciphersuites as defined in NDcPP30e:FCS_TLSS_EXT.1.1.

**NDcPP30e:FCS_TLSS_EXT.1.6**

The TSF shall prohibit the use of the following extensions:

- Early data extension

**NDcPP30e:FCS_TLSS_EXT.1.7**

The TSF shall [***not use PSKs***].

**NDcPP30e:FCS_TLSS_EXT.1.8**

The TSF shall [***support secure renegotiation in accordance with RFC 5746 by always including the "renegotiation_info" TLS extension in TLS 1.2 ServerHello messages, reject [TLS 1.2] renegotiation attempts***].

***Application Note:*** *ECDHE curves secp256r1 and secp521r1 are only supported for the WebUI. For FCS_TLSC_EXT.1.8 the SMC distributed TOE channel supports secure renegotiation, while the SMC Web UI and Engine distributed TOE channel reject renegotiation attempts.*

#### 5.1.3.18 TLS Server Support for Mutual Authentication (NDcPP30e:FCS_TLSS_EXT.2)

**NDcPP30e:FCS_TLSS_EXT.2.1**

The TSF shall support TLS communication with mutual authentication of TLS clients using X.509v3 certificates and shall [

- ***reject the connection if the client either does not provide a client certificate at all or the client certificate cannot be successfully validated by the TOE (except for override mechanisms that might be defined in NDcPP30e:FCS_TLSS_EXT.2.2) ('hard fail')***

].

**NDcPP30e:FCS_TLSS_EXT.2.2**

When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [***not implement any administrator override mechanism***].

**NDcPP30e:FCS_TLSS_EXT.2.3**

The TSF shall not establish a trusted channel if the identifier contained in a certificate does not match an expected identifier for the client. If the identifier is a Fully Qualified Domain Name (FQDN), then the TSF shall match the identifiers according to RFC 6125, otherwise the TSF shall parse the identifier from the certificate and match the identifier against the expected identifier of the client as described in the TSS.

**NDcPP30e:FCS_TLSS_EXT.2.4**

The TSF shall present a [***TLS 1.2***] Certificate Request message containing the following algorithms: [

**FPT_ITT (Engine as Server)**
***ecdsa_secp256r1 with sha256(0x0403),***
***ecdsa_secp384r1 with sha384(0x0503),***
***ecdsa_secp521r1 with sha512(0x0603)***

**FPT_ITT (SMC as Server)**
***ecdsa_secp384r1 with sha384(0x0503)***

] and no other algorithms.

## 5.1.4 User data protection (FDP)

### 5.1.4.1 Full Residual Information Protection (STFFW14e:FDP_RIP.2)

**STFFW14e: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.

## 5.1.5 Firewall (FFW)

### 5.1.5.1 Stateful Traffic Filtering (STFFW14e:FFW_RUL_EXT.1)

**STFFW14e:FFW_RUL_EXT.1.1**

The TSF shall perform stateful traffic filtering on network packets processed by the TOE.

**STFFW14e: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.

**STFFW14e: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.

**STFFW14e:FFW_RUL_EXT.1.4**

The TSF shall allow the stateful traffic filtering rules to be assigned to each distinct network interface.

**STFFW14e: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***].

**STFFW14e: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;
e) 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;
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 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;
g) 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;
h) 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
i) [***no other rules***].

**STFFW14e: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.

**STFFW14e:FFW_RUL_EXT.1.8**

The TSF shall process the applicable stateful traffic filtering rules in an administratively defined order.

**STFFW14e:FFW_RUL_EXT.1.9**

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

**STFFW14e: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***].

#### 5.1.5.2 Stateful Filtering of Dynamic Protocols (STFFW14e:FFW_RUL_EXT.2)

**STFFW14e: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***].

### 5.1.6 Identification and authentication (FIA)

#### 5.1.6.1 Authentication Failure Management (NDcPP30e:FIA_AFL.1)

**NDcPP30e:FIA_AFL.1.1**

The TSF shall detect when an Administrator configurable positive integer within [**1-1000**]

unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

**NDcPP30e:FIA_AFL.1.2**

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

#### 5.1.6.2 Password Management (NDcPP30e:FIA_PMG_EXT.1)

**NDcPP30e:FIA_PMG_EXT.1.1**

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

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

b) Minimum password length shall be configurable to between [**1**] and [**80**] characters.

#### 5.1.6.3 Pre-Shared Key Composition (VPNGW13:FIA_PSK_EXT.1)

**VPNGW13:FIA_PSK_EXT.1.1**

The TSF shall be able to use pre-shared keys for IPsec and [***IKEv2***].

**VPNGW13:FIA_PSK_EXT.1.2**

The TSF shall be able to accept the following as pre-shared keys: [***generated bit-based***] keys.

#### 5.1.6.4 Generated Pre-Shared Keys (VPNGW13:FIA_PSK_EXT.2)

**VPNGW13:FIA_PSK_EXT.2.1**

The TSF shall be able to [***accept externally generated pre-shared keys***].

#### 5.1.6.5 Protected Authentication Feedback (NDcPP30e:FIA_UAU.7)

**NDcPP30e: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.

#### 5.1.6.6 User Identification and Authentication - per TD0900 (NDcPP30e:FIA_UIA_EXT.1)

**NDcPP30e: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]***].

**NDcPP30e: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.

**NDcPP30e:FIA_UIA_EXT.1.3**

The TSF shall provide the following remote authentication mechanisms [***Web GUI password***] and [***no other mechanism***]. The TSF shall provide the following local authentication mechanisms [***password-based***]. (TD0900 applied)

**NDcPP30e:FIA_UIA_EXT.1.4**

The TSF shall authenticate any administrative user's claimed identity according to each authentication mechanism specified in NDcPP30e:FIA_UIA_EXT.1.3.

#### 5.1.6.7 X.509 Certificate Validation (NDcPP30e:FIA_X509_EXT.1/ITT)

**NDcPP30e:FIA_X509_EXT.1.1/ITT**

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of two certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***no revocation method***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Server certificates presented for DTLS/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 DTLS/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.

**NDcPP30e:FIA_X509_EXT.1.2/ITT**

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.

#### 5.1.6.8 X.509 Certificate Validation (NDcPP30e:FIA_X509_EXT.1/Rev)

**NDcPP30e:FIA_X509_EXT.1.1/Rev**

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5***]
- 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 DTLS/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 DTLS/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.

**NDcPP30e:FIA_X509_EXT.1.2/Rev**

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

#### 5.1.6.9 X.509 Certificate Validation (VPNGW13:FIA_X509_EXT.1/Rev)

**VPNGW13:FIA_X509_EXT.1.1/Rev**

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.

- The TSF shall validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5***].
- 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.

#### 5.1.6.10 X.509 Certificate Authentication (NDcPP30e:FIA_X509_EXT.2)

**NDcPP30e:FIA_X509_EXT.2.1**
The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [***IPsec, TLS***], and [***no additional uses***].

**NDcPP30e:FIA_X509_EXT.2.2**
When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [***not accept the certificate***].

#### 5.1.6.11 X.509 Certificate Authentication (VPNGW13:FIA_X509_EXT.2)

**VPNGW13:FIA_X509_EXT.2.1**
The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec and [***TLS***], and [***no additional uses***].

**VPNGW13:FIA_X509_EXT.2.2**
When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall: [***not accept the certificate***].

#### 5.1.6.12 X.509 Certificate Requests (NDcPP30e:FIA_X509_EXT.3)

**NDcPP30e:FIA_X509_EXT.3.1**
The TSF shall generate a Certification Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [***device-specific information, Common Name, Organization, Organizational Unit, Country***].

**NDcPP30e:FIA_X509_EXT.3.2**
The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

#### 5.1.6.13 X.509 Certificate Requests (VPNGW13:FIA_X509_EXT.3)

**VPNGW13:FIA_X509_EXT.3.1**
The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [***device-specific information, Common Name, Organization, Organizational Unit, Country***].

**VPNGW13:FIA_X509_EXT.3.2**
The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

## 5.1.7 Security management (FMT)

### 5.1.7.1 Management of Security Functions Behaviour (NDcPP30e:FMT_MOF.1/Functions)

**NDcPP30e:FMT_MOF.1.1/Functions**

The TSF shall restrict the ability to [***determine the behaviour of, modify the behaviour of***] the functions [***transmission of audit data to an external IT entity, handling of audit data, audit functionality when Local Audit Storage Space is full***] to Security Administrators.

### 5.1.7.2 Management of security functions behaviour (NDcPP30e:FMT_MOF.1/ManualUpdate)

**NDcPP30e:FMT_MOF.1.1/ManualUpdate**

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

### 5.1.7.3 Management of TSF Data (NDcPP30e:FMT_MTD.1/CoreData)

**NDcPP30e:FMT_MTD.1.1/CoreData**

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

### 5.1.7.4 Management of TSF Data (NDcPP30e:FMT_MTD.1/CryptoKeys)

**NDcPP30e:FMT_MTD.1.1/CryptoKeys**

The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

### 5.1.7.5 Management of TSF Data (VPNGW13:FMT_MTD.1/CryptoKeys)

**VPNGW13:FMT_MTD.1.1/CryptoKeys**

The TSF shall restrict the ability to manage the cryptographic keys and certificates used for VPN operation to Security Administrators.

### 5.1.7.6 Specification of Management Functions - per TD0880 (NDcPP30e:FMT_SMF.1)

**NDcPP30e:FMT_SMF.1.1**

The TSF shall be capable of performing the following management functions:

- Ability to administer the TOE remotely;
- Ability to configure the access banner;
- Ability to configure the remote session inactivity time before session termination;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- [***Ability to modify the behavior of the transmission of audit data to an external IT entity,***
- ***Ability to manage the cryptographic keys,***
- ***Ability to configure the cryptographic functionality,***
- ***Ability to configure the lifetime for IPsec SAs,***
- ***Ability to configure the list of supported (D)TLS ciphers,***
- ***Ability to configure the interaction between TOE components,***
- ***Ability to set the time which is used for time-stamps,***
- ***Ability to configure NTP,***
- ***Ability to configure the reference identifier for the peer,***
- ***Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors,***
- ***Ability to generate Certificate Signing Request (CSR) and process CA certificate response,***
- ***Ability to administer the TOE locally,***
- ***Ability to configure the local session inactivity time before session termination or locking,***
- ***Ability to configure the authentication failure parameters for FIA_AFL.1***

]. (TD0880 applied)

#### 5.1.7.7 Specification of Management Functions (STFFW14e:FMT_SMF.1/FFW)

**STFFW14e:FMT_SMF.1.1/FFW**

The TSF shall be capable of performing the following management functions: Ability to configure firewall rules.

#### 5.1.7.8 Specification of Management Functions (VPNGW13:FMT_SMF.1/VPN)

**VPNGW13:FMT_SMF.1.1/VPN**

The TSF shall be capable of performing the following management functions:
- Definition of packet filtering rules
- Association of packet filtering rules to network interfaces
- Ordering of packet filtering rules by priority
- [***Configuration of remote VPN client session timeout,***
- ***Configuration of attributes used to deny establishment of remote VPN client session***].

#### 5.1.7.9 Restrictions on Security Roles (NDcPP30e:FMT_SMR.2)

**NDcPP30e:FMT_SMR.2.1**

The TSF shall maintain the roles:
- Security Administrator.

**NDcPP30e:FMT_SMR.2.2**

The TSF shall be able to associate users with roles.

**NDcPP30e:FMT_SMR.2.3**

The TSF shall ensure that the conditions
- The Security Administrator role shall be able to administer the TOE remotely

are satisfied.

### 5.1.8 Packet Filtering (FPF)

#### 5.1.8.1 Packet Filtering Rules (VPNGW13:FPF_RUL_EXT.1)

**VPNGW13:FPF_RUL_EXT.1.1**

The TSF shall perform Packet Filtering on network packets processed by the TOE.

**VPNGW13:FPF_RUL_EXT.1.2**

The TSF shall allow the definition of packet filtering rules using the following network protocols and protocol fields:
- IPv4 (RFC 791)
  - Source address
  - Destination address
  - Protocol
- IPv6 (RFC 8200)
  - Source address
  - Destination address
  - Protocol
- TCP (RFC 793)
  - Source port
  - Destination port
- UDP (RFC 768)
  - Source port
  - Destination port

**VPNGW13:FPF_RUL_EXT.1.3**

The TSF shall allow the following operations to be associated with packet filtering rules: permit and drop with the capability to log the operation.

**VPNGW13:FPF_RUL_EXT.1.4**

The TSF shall allow the packet filtering rules to be assigned to each distinct network interface.

**VPNGW13:FPF_RUL_EXT.1.5**
The TSF shall process the applicable packet filtering rules (as determined in accordance with VPNGW13:FPF_RUL_EXT.1.4) in the following order: Administrator-defined.
**VPNGW13:FPF_RUL_EXT.1.6**
The TSF shall drop traffic if a matching rule is not identified.

## 5.1.9 Protection of the TSF (FPT)

### 5.1.9.1 Protection of Administrator Passwords (NDcPP30e:FPT_APW_EXT.1)

**NDcPP30e:FPT_APW_EXT.1.1**
The TSF shall store administrative passwords in non-plaintext form.
**NDcPP30e:FPT_APW_EXT.1.2**
The TSF shall prevent the reading of plaintext administrative passwords.

### 5.1.9.2 Failure with Preservation of Secure State (Self-Test Failures) (VPNGW13:FPT_FLS.1/SelfTest)

**VPNGW13:FPT_FLS.1.1/SelfTest**
The TSF shall shut down when the following types of failures occur: failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests.

### 5.1.9.3 Basic internal TSF data transfer protection (NDcPP30e:FPT_ITT.1)

**NDcPP30e:FPT_ITT.1.1**
The TSF shall protect TSF data from disclosure and detect its modification when it is transmitted between separate parts of the TOE through the use of [***TLS***].

### 5.1.9.4 Protection of TSF Data (for reading of all symmetric keys) (NDcPP30e:FPT_SKP_EXT.1)

**NDcPP30e:FPT_SKP_EXT.1.1**
The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

### 5.1.9.5 Reliable Time Stamps (NDcPP30e:FPT_STM_EXT.1)

**NDcPP30e:FPT_STM_EXT.1.1**
The TSF shall be able to provide reliable time stamps for its own use.
**NDcPP30e:FPT_STM_EXT.1.2**
The TSF shall [***allow the Security Administrator to set the time, synchronize time with an NTP server***].

### 5.1.9.6 TSF testing - per TD0836 (NDcPP30e:FPT_TST_EXT.1)

**NDcPP30e:FPT_TST_EXT.1.1**
The TSF shall run a suite of the following self-tests:
- During initial start-up (on power on) to verify the integrity of the TOE firmware and software
- Prior to providing any cryptographic service and [***at no other time***] to verify correct operation of cryptographic implementation necessary to fulfil the TSF
- [***no other***] self-tests [**beyond those listed in the two bullets above**]

to demonstrate the correct operation of the TSF. (TD0836 applied)
**NDcPP30e:FPT_TST_EXT.1.2**
The TSF shall respond to [***all failures***] by [***rebooting(Engine), [halting (SMC)]***].

### 5.1.9.7 TSF Testing - per TD0824 (VPNGW13:FPT_TST_EXT.1)

**VPNGW13:FPT_TST_EXT.1.1**
The TSF shall run a suite of the following self-tests:

- During initial start-up (on power on) to verify the integrity of the TOE firmware and software;
- Prior to providing any cryptographic service and [***at no other time***] to verify correct operation of cryptographic implementation necessary to fulfil the TSF;
- [***no other***] self tests [**beyond those listed in the two bullets above**]

to demonstrate the correct operation of the TSF: noise source health tests. (TD0824 applied)

**VPNGW13:FPT_TST_EXT.1.2**

The TSF shall respond to [***all failures***] by [***rebooting(Engine), [halting (SMC)]***] (per TD0824)

#### 5.1.9.8 Self-Test with Defined Methods (VPNGW13:FPT_TST_EXT.3)

**VPNGW13:FPT_TST_EXT.3.1**

The TSF shall run a suite of the following self-tests when loaded for execution to demonstrate the correct operation of the TSF: integrity verification of stored executable code.

**VPNGW13:FPT_TST_EXT.3.2**

The TSF shall execute the self-testing through a TSF-provided cryptographic service specified in FCS_COP.1/SigGen.

#### 5.1.9.9 Trusted update (NDcPP30e:FPT_TUD_EXT.1)

**NDcPP30e:FPT_TUD_EXT.1.1**

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

**NDcPP30e: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***].

**NDcPP30e:FPT_TUD_EXT.1.3**

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

#### 5.1.9.10 Trusted Update - per TD0824 (VPNGW13:FPT_TUD_EXT.1)

**VPNGW13:FPT_TUD_EXT.1.1**

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

**VPNGW13: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***].

**VPNGW13:FPT_TUD_EXT.1.3**

The TSF shall provide means to authenticate firmware/software updates to the TOE using a digital signature mechanism and [***no other mechanisms***] prior to installing those updates. (TD0824 applied)

### 5.1.10 TOE access (FTA)

#### 5.1.10.1 TSF-initiated Termination (NDcPP30e:FTA_SSL.3)

**NDcPP30e:FTA_SSL.3.1**

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

#### 5.1.10.2 TSF-Initiated Termination (VPN Headend) (VPNGW13:FTA_SSL.3/VPN)

**VPNGW13:FTA_SSL.3.1/VPN**

The TSF shall terminate a remote VPN client session after an Administrator-configurable time interval of session inactivity.

#### 5.1.10.3 User-initiated Termination (NDcPP30e:FTA_SSL.4)

**NDcPP30e:FTA_SSL.4.1**

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

#### 5.1.10.4 TSF-initiated Session Locking (NDcPP30e:FTA_SSL_EXT.1)

**NDcPP30e:FTA_SSL_EXT.1.1**

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

#### 5.1.10.5 Default TOE Access Banners (NDcPP30e:FTA_TAB.1)

**NDcPP30e:FTA_TAB.1.1**

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

#### 5.1.10.6 TOE Session Establishment (VPNGW13:FTA_TSE.1)

**VPNGW13:FTA_TSE.1.1**

The TSF shall be able to deny session establishment of a remote VPN client session based on location, time, day, [***no other attributes***].

#### 5.1.10.7 VPN Client Management (VPNGW13:FTA_VCM_EXT.1)

**VPNGW13:FTA_VCM_EXT.1.1**

The TSF shall assign a private IP address to a VPN client upon successful establishment of a security session.

### 5.1.11 Trusted path/channels (FTP)

#### 5.1.11.1 Inter-TSF trusted channel (NDcPP30e:FTP_ITC.1)

**NDcPP30e: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.

**NDcPP30e:FTP_ITC.1.2**

The TSF shall permit [***the TSF***] to initiate communication via the trusted channel.

**NDcPP30e:FTP_ITC.1.3**

The TSF shall initiate communication via the trusted channel for [**transmitting audit records to a remote audit server**].

#### 5.1.11.2 Inter-TSF Trusted Channel (VPN Communications) (VPNGW13:FTP_ITC.1/VPN)

**VPNGW13:FTP_ITC.1.1/VPN**

The TSF shall be capable of using IPsec to provide a communication channel between itself and authorized IT entities supporting VPN communications that is logically distinct from other

communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

**VPNGW13:FTP_ITC.1.2/VPN**

The TSF shall permit the authorized IT entities to initiate communication via the trusted channel.

**VPNGW13:FTP_ITC.1.3/VPN**

The TSF shall initiate communication via the trusted channel for: [***remote VPN gateways or peers***].

#### 5.1.11.3 Trusted Path (NDcPP30e:FTP_TRP.1/Admin)

**NDcPP30e:FTP_TRP.1.1/Admin**

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

**NDcPP30e:FTP_TRP.1.2/Admin**

The TSF shall permit remote Administrators to initiate communication via the trusted path.

**NDcPP30e:FTP_TRP.1.3/Admin**

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

#### 5.1.11.4 Trusted Path (NDcPP30e:FTP_TRP.1/Join)

**NDcPP30e:FTP_TRP.1.1/Join**

The TSF shall provide a communication path between itself and a joining component that is logically distinct from other communication paths and provides assured identification of [***both joining component and TSF endpoint***] and protection of the communicated data from modification and [***disclosure***].

**NDcPP30e:FTP_TRP.1.2/Join**

The TSF shall permit [***the joining component***] to initiate communication via the trusted path.

**NDcPP30e:FTP_TRP.1.3/Join**

The TSF shall require the use of the trusted path for joining components to the TSF under environmental constraints identified in [**the Forcepoint Network Security Platform 7.3 Common Criteria Evaluated Configuration Guide**].

## 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
- Communication
- Cryptographic support
- User data protection
- Firewall
- Identification and authentication
- Security management
- Packet Filtering
- 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 Security Engine generates audit events pertaining to packet filtering. All other security relevant audit events are generated by all TOE components.

The Security Engine transfers audit records to the Virtual SMC Appliance Log Server immediately after generation of the record. The Virtual 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 Virtual 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.

The Log Server, which aggregates audit records from Security Engines, writes incoming audit entries to the Virtual 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 Security Engine and the Virtual 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. 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 Virtual 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 GUI), and when less than 100MB of space remains, the Management Server will no longer allow the administrator to make configuration changes (as such a change would result in an audit message that the Management Server no longer has sufficient space to safely store) until action is taken by the administrator to delete local audit data.

The Security audit function satisfies the following security functional requirements:

- NDcPP30e/STFFW14e: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 time and date, admin name, subject (key/element name), operation (generation/deletion), result. The TOE also inserts into audit records all of the additional information described by **Table 2 Audit events**.
- NDcPP30e:FAU_GEN.2: Every audit record indicates the SMC user responsible for the action. Additionally, the TOE records the key name (the string provided by the administrator) as an identifier of any TOE key generated, imported, changed, or deleted.
- NDcPP30e:FAU_GEN_EXT.1: The Security Engines and the Virtual SMC Appliance each generate audit records specific to the component's functionality. Security Engines primarily generate firewall and TLS secure channel audits while the Virtual SMC Appliance generates TLS secure channel audits and audits of administrative actions.
- NDcPP30e:FAU_STG_EXT.1/4/5: All audit records generated by the Security Engine are transmitted first to the Virtual SMC Appliance's Log Server and then (if configured by an administrator) forwarded to an external syslog server using TLS protected syslog. The SMC can forward its audit records to an external syslog server using TLS protected syslog (if configured by an administrator). Virtual SMC Appliance, should it exhaust its audit log storage space, halts generation of new audit data.
- VPNGW13:FAU_GEN.1/VPN: The Engine components create all VPN (IKE and IPsec) related audit events defined in **Table 2 Audit events**. (i.e., all audit events beginning with "VPNGW13"), and the Engines use the same audit mechanisms for VPN audits as for all other audits.

## 6.2 Communication

The Communication function satisfies the following security functional requirements:

- NDcPP30e:FCO_CPC_EXT.1: The TOE implements a registration process using a channel that meets the secure registration channel requirements in FTP_TRP.1/Join. The TOE utilizes TLS to provide a registration channel between a new Engine and the SMC. The TOE requires that an administrator first create a new Security Engine object from within the SMC. Upon completion, the SMC generates and provides the administrator with a 45-character "password" (the SMC generates a unique "password" for each and every registration attempt) as well as the SHA-512 hash of the SMC's server TLS certificate. The administrator then inputs this password into the Engine (via console) and initiates the registration connection from the Engine. The Engine confirms that the SMC's TLS certificate SHA-512 hash either matches the administrator pre-configured value (if the administrator chose to enter it) or displays the hash for the administrator to confirm. Then the SMC authenticates the Engine by requesting the Engine send the SHA-512 hash of the SMC-generated password. Once authenticated, the SMC pushes the SMC's internal CA certificate to the Engine, the Engine creates a CSR that the SMC's internal CA uses to issue the Engine a TLS certificate, and the Engine validates the received TLS certificate. SMC then pushes a policy to the Engine. The policy contains the Log Server reference identifier.

After completion, the Engine and SMC subsequently communicate through mutually-authenticated TLS connections.

An administrator can disable communication between an Security Engine and the SMC via the SMC Client GUI. This is done from the perspective of the Security Engine by performing a factory reset of the Engine and from the perspective of the SMC by deleting the Security Engine element in the Client GUI.

Finally, note that the TOE, by design, restricts communications between components and only permits communications between the SMC and Engines, and does not support direct communication between two Engines.

## 6.3 Cryptographic support

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

1. Security Engine uses cryptography from
    a. the Forcepoint Network Security Platform Security Engine FIPS Cryptographic Module 2.0 (based upon SafeZone FIPS Cryptographic Module 2.0) for IKE connections,
    b. the Forcepoint Network Security Platform Security Engine FIPS Library 3.1.2 (based upon OpenSSL 3.1.2 FIPS) for TLS connections, code integrity, and trusted updates,
    c. the Forcepoint Network Security Platform Security Engine Cryptographic Kernel Module 3.0 for IPsec connections.
2. Virtual SMC Appliance uses cryptography from
    a. the Forcepoint Network Security Platform SMC FIPS Library 3.0.7 (based on Oracle Linux 9 OpenSSL FIPS Provider 3.0.7-b27cdeb3ba51be46) for password hashing, integrity testing and verification of Trusted Updates,
    b. the Forcepoint Network Security Platform SMC FIPS Java API 2.0.1 for TLS, CSR generation, and password hashing for its Management and Log Server,
    c. the Forcepoint Network Security Platform SMC FIPS Cryptographic Module for NTP 3.101 for its NTP daemon.

Version 7.3 of the Security Engine internally includes version 12 of the Forcepoint Network Security Platform OS, which itself includes a Linux 6.6 kernel.

| SFR | Algorithm | NIST/ISO Standard | Cert# |
|---|---|---|---|
| FCS_CKM.1 (Key Gen)<br>FCS_CKM.1/IKE | RSA IFC Key Generation<br>2048/3072/4096 bit | FIPS 186-5, RSA | A8030 |
| | ECDSA ECC Key Generation<br>P-256, P-384, P-521 | FIPS 186-5, ECDSA | A8030 |
| | FFC Schemes using 'safe-prime' groups DH-14, DH-15, DH-16, DH-17, DH-18 | NIST SP 800-56A | Tested with a known good implementation |
| FCS_CKM.2 (Key Establishment) | ECC-based Key Exchange<br>P-256, P-384, P-521 | SP 800-56A,<br>KAS ECC | A8030 |
| | FFC Schemes using 'safe-prime' groups DH-14, DH-15, DH-16, DH-17, DH-18 | NIST SP 800-56A | Tested with a known good implementation |
| FCS_COP.1/DataEncryption | AES CBC - 128/192/256<br>AES GCM - 128/192/256 | ISO 10116<br>ISO 19772 | A8030 |
| FCS_COP.1/SigGen | ECDSA Sign/Verify<br>P-256, P-384, P-521 | FIPS 186-5, ECDSA | A8030 |
| | RSA Sign/Verify<br>2048, 3072, 4096 | FIPS 186-5, RSA | A8030 |

| SFR | Algorithm | NIST/ISO Standard | Cert# |
|---|---|---|---|
| FCS_COP.1/Hash | SHA Hashing<br>SHA-1, SHA-256, SHA-384, SHA-512 | ISO/IEC 10118-3:2004 | A8030 |
| FCS_COP.1/KeyedHash | Keyed Hash<br>HMAC-SHA-1<br>HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-521 | ISO/IEC 9797-2:2011 | A8030 |
| FCS_RBG_EXT.1 (CTR) (Random) | CTR_DRBG (AES)<br>256 bits | ISO/IEC 18031:2011 | A8030 |

**Table 4 Forcepoint Network Security Platform Security Engine FIPS Cryptographic Module 2.0 (based upon SafeZone FIPS Cryptographic Module 2.0) CAVP Certificates**

| SFR | Algorithm | NIST/ISO Standard | Cert# |
|---|---|---|---|
| FCS_CKM.1 (Key Gen) | RSA IFC Key Generation<br>2048/3072/4096 bit | FIPS 186-5, RSA | A8029 |
| | ECDSA ECC Key Generation<br>P-256, P-384, P-521 | FIPS 186-5, ECDSA | A8029 |
| FCS_CKM.2 (Key Establishment) | ECC-based Key Exchange<br>P-256, P-384, P-521 | SP 800-56A,<br>KAS ECC | A8029 |
| FCS_COP.1/DataEncryption | AES CBC - 128/256<br>AES GCM - 128/256 | ISO 10116<br>ISO 19772 | A8029 |
| FCS_COP.1/SigGen | ECDSA Sign/Verify<br>P-256, P-384, P-521 | FIPS 186-5, ECDSA | A8029 |
| | RSA Sign/Verify<br>2048, 3072, 4096 | FIPS 186-5, RSA | A8029 |
| FCS_COP.1/Hash | SHA Hashing<br>SHA-1, SHA-256, SHA-384, SHA-512 | ISO/IEC 10118-3:2004 | A8029 |
| FCS_COP.1/KeyedHash | Keyed Hash<br>HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 | ISO/IEC 9797-2:2011 | A8029 |
| FCS_RBG_EXT.1 (CTR) (Random) | CTR_DRBG (AES)<br>256 bits | ISO/IEC 18031:2011 | A8029 |

**Table 5 Forcepoint Network Security Platform Security Engine FIPS Library 3.1.2 (based upon OpenSSL 3.1.2 FIPS) CAVP Certificates**

| SFR | Algorithm | NIST/ISO Standard | Cert# |
|---|---|---|---|
| FCS_COP.1/DataEncryption | AES 128/192/256 CBC, GCM | ISO 10116<br>ISO 19772 | A7802 |
| FCS_COP.1/Hash | SHA Hashing<br>SHA-1, SHA-256, SHA-384, SHA-512 | ISO/IEC 10118-3:2004 | A7802 |
| FCS_COP.1/KeyedHash | Keyed Hash<br>HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512 | ISO/IEC 9797-2:2011 | A7802 |

**Table 6 Forcepoint Network Security Platform Security Engine Cryptographic Kernel Module 3.0 CAVP Certificates**

The TOE includes the Forcepoint Network Security Platform SMC FIPS Java API 2.0.1

| SFR | Algorithm | NIST Standard | Cert# |
|---|---|---|---|
| FCS_CKM.1 (Key Gen) | RSA IFC Key Generation<br>2048/3072 bit | FIPS 186-5, RSA | A8085 |

| SFR | Algorithm | NIST Standard | Cert# |
|---|---|---|---|
| | ECDSA ECC Key Generation<br>P-256, P-384, P-521 | FIPS 186-5, ECDSA | A8085 |
| | FFC-based Key Generation<br>2048/3072 bit | FIPS PUB 186-4, FFC | A8085 |
| FCS_CKM.2 (Key Establishment) | ECC-based Key Exchange<br>P-256, P-384, P-521 | SP 800-56A,<br>KAS ECC | A8085 |
| | FFC-based Key Exchange<br>2048/3072 bit | FIPS PUB 186-4, FFC | A8085 |
| FCS_COP.1/DataEncryption | AES 128/256 CBC, GCM | ISO 10116<br>ISO 19772 | A8085 |
| FCS_COP.1/SigGen | RSA Sign/Verify<br>2048/3072 bit | FIPS 186-5, RSA | A8085 |
| | ECDSA Sign/Verify<br>P-256, P-384, P-521 | FIPS 186-5, ECDSA | A8085 |
| FCS_COP.1/Hash | SHA Hashing<br>SHA-1, SHA-256, SHA-384, SHA-512 | ISO/IEC 10118-3:2004 | A8085 |
| FCS_COP.1/KeyedHash | Keyed Hash<br>HMAC-SHA-1, HMAC-SHA-256,<br>HMAC-SHA-384, HMAC-SHA-521 | ISO/IEC 9797-2:2011 | A8085 |
| FCS_RBG_EXT.1 (Hash)(Random) | DRBG Bit Generation<br>Hash_DRBG (SHA-512) | ISO/IEC 18031:2011 | A8085 |

**Table 7 Virtual SMC Appliance Forcepoint Network Security Platform SMC FIPS Java API 2.0.1 CAVP Certificates**

Version 7.3 of the Virtual SMC Appliance internally includes a VMWare ESXi 8.0 hypervisor, running a Oracle Linux 9 with RHCK virtual machine. The Oracle Linux 9 with RHCK OS (running as a virtual machine) includes a Linux 5.14 kernel (RHCK).

| SFR | Algorithm | NIST Standard | Cert# |
|---|---|---|---|
| FCS_COP.1/SigGen | ECDSA Sign/Verify<br>P-256, P-384, P-521 | FIPS 186-5, ECDSA | A8086 |
| FCS_COP.1/Hash | SHA Hashing<br>SHA-1, SHA-256, SHA-384, SHA-512 | ISO/IEC 10118-3:2004 | A8086 |
| FCS_COP.1/KeyedHash | Keyed Hash<br>HMAC-SHA-256, HMAC-SHA-384 | ISO/IEC 9797-2:2011 | A8086 |

**Table 8 Virtual SMC Appliance Forcepoint Network Security Platform SMC FIPS Library 3.0.7 (based on Oracle Linux 9 OpenSSL FIPS Provider 3.0.7-b27cdeb3ba51be46) CAVP Certificates**

| SFR | Algorithm | NIST Standard | Cert# |
|---|---|---|---|
| FCS_COP.1/Hash | SHA Hashing, SHA-1 | ISO/IEC 10118-3:2004 | A7836 |

**Table 9 Virtual SMC Appliance Forcepoint Network Security Platform SMC FIPS Cryptographic Module for NTP 3.101 CAVP Certificates**

### 6.3.1 Security Engine

The Security Engine directly calls the Forcepoint Network Security Platform Security Engine FIPS Cryptographic Module 2.0 (based upon SafeZone FIPS Cryptographic Module 2.0) for hashing, HMAC, encryption/decryption, and signature related operations supporting IKE connections (again see **Table 4**).

The Engine also uses the Forcepoint Network Security Platform Security Engine FIPS Library 3.1.2 (based upon OpenSSL 3.1.2 FIPS) library for

- TLS connections (hashing, HMAC, encryption/decryption, and signature related operations),
- CSR generation,
- Engine integrity testing, and
- Engine Trusted Updates.

Finally, the Security Engine uses the Forcepoint Network Security Platform Security Engine Cryptographic Kernel Module 3.0 to provide hashing, HMAC, and encryption/decryption operations supporting IPsec/ESP connections (see **Table 6**).

### 6.3.2 Virtual SMC Appliance

The primary functions of the Virtual SMC Appliance are provided by the Management Server and the Log Server (described in following sections). However, the Virtual 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 6.9 Protection of the TSF, and the cryptography verifying the validity of the update comes from the Virtual SMC Appliance Forcepoint Network Security Platform SMC FIPS Library 3.0.7 (based on Oracle Linux 9 OpenSSL FIPS Provider 3.0.7-b27cdeb3ba51be46).

The Virtual SMC Appliance Forcepoint Network Security Platform SMC FIPS Library 3.0.7 (based on Oracle Linux 9 OpenSSL FIPS Provider 3.0.7-b27cdeb3ba51be46) uses cryptography for (local console) password hashing, for power-up integrity testing, and for signature verification of TOE updates.

The Virtual SMC Appliance uses a software noise source and uses a platform-based noise source. The Virtual SMC Appliance uses these noise sources to instantiate its Forcepoint Network Security Platform SMC FIPS Java API 2.0.1's SHA-512 HASH_DRBG and generate keys. The Forcepoint Network Security Platform SMC FIPS Java API 2.0.1 is used by the Management Server and Log Server as described below.

#### 6.3.2.1 Management Server

The Virtual 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 Network Security Platform SMC FIPS Java API 2.0.1 for all encryption, decryption, hashing and signature operations associated with support for the TLS protocol. That Java library draws from /dev/random to instantiate its DRBG (a SHA-512 Hash_DRBG).

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 10.

The Management Server also accepts incoming administrative sessions (SMC Client connections) that are protected by TLS, and uses the cipher suites identified by Table 11 below.

#### 6.3.2.2 Log Server

The Virtual SMC Appliance's Log Server communicates with the Security 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 Network Security Platform SMC FIPS Java API 2.0.1 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_DHE_RSA_WITH_AES_128_CBC_SHA256 |
| TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 |
| TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 |
| TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 |
| TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 |
| TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 |
| TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 |
| TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 |
| TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 |
| TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 |
| TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 |
| TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 |

**Table 10 Cipher suites to communicate with an External Syslog Server**

The Forcepoint Network Security Platform SMC FIPS Java API 2.0.1 Library used by the Virtual SMC Appliance (and thus used by both the Management and Log Servers) uses the Forcepoint Entropy Source Frontend library (libentropy.so.1.5) to instantiate its Forcepoint Network Security Platform SMC FIPS Java API 2.0.1's SHA-512 HASH_DRBG and generate keys and that is described as part of the Virtual SMC Appliance.

The Log Server also accepts incoming administrative sessions (SMC Client connections) that are protected by TLS and authorized by the Management Server. The Log Server and Management Server utilize the following cipher suites to protect remote administrative sessions.

| |
|---|
| TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 |
| TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 |

**Table 11 Cipher suites to communicate with remote administrators**

For all inter-TOE communication, the following cipher is used by both the SMC and the Security Engine to protect the communication between the distributed TOE components.

| |
|---|
| TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 |

**Table 12 Cipher suite for distributed TOE communication**

Note that the ciphersuite identified in Table 12 Cipher suite for distributed TOE communication above is the only one supported by both the SMC and the Security Engine for distributed TOE communication.

### 6.3.3 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 |
| IKEv2 RSA or ECDSA authentication private key | On Disk | Command | Overwriting with pseudo-random data |
| IKEv2 SA session keys | In Memory | SA deleted | Overwriting with zeros |

| CSP or Key: | Stored in | Zeroized upon: | Zeroized by: |
|---|---|---|---|
| ESP SA session keys | In Memory | SA deleted | Overwriting with zeros |
| Passwords | On Disk | Command | Overwriting once with pseudo-random data |

**Table 13 CSPs and Keys**

The Cryptographic support function satisfies the following security functional requirements:

- NDcPP30e:FCS_CKM.1: The Virtual 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, FFC or ECDH keys used during TLS cipher suite negotiation.

| TOE Component | Client or Server? | FFC key gen? | ECDH gen? | ECDSA gen? | RSA gen? |
|---|---|---|---|---|---|
| Mgmt Server | Both | Yes | Yes | Yes | Yes |
| Log Server | Both | Yes | Yes | Yes | Yes |
| Engine | Both | N/A | Yes | Yes | No |

  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).

- VPNGW13:FCS_CKM.1/IKE: The Engines support asymmetric key generation for IKE Authentication. The following table details which components act as IKE peers and can generate RSA, ECDSA, or FFC safe primes keys (in accordance with FIPS PUB 186-5) used during IKEv2 negotiation.

| TOE Component | Performs IKE? | ECDSA gen? | RSA gen? | FFC safe primes gen? |
|---|---|---|---|---|
| Mgmt Server | No | No | No | No |
| Log Server | No | No | No | No |
| Engine | Yes | Yes | Yes | Yes |

  The TOE permits the administrator to choose the type and key strength (size) of the authentication key pair to generate or import, including ECDSA [P-256, P-384, or P-521] or RSA [2048, 3072, 4096].

  The TOE performs RSA key generation as per FIPS 186-5 Appendix A.1 (IFC Key Pair Generation) and ECDSA key generation per FIPS 186-5 Appendix A.2 (ECC Key Pair Generation). The TOE's implementation complies with the appendices, implementing all shall requirements (omitting none) and does not implement any additional ('shall not', 'should', 'should not', nor TOE-specific extensions) functionality

- NDcPP30e:FCS_CKM.2: the following table identifies the supported key establishment schemes mapped to the associated SFRs and their usage.

| Scheme | SFR | Services |
|---|---|---|
| FFC (2048 or 3072 bits) | FCS_TLSC_EXT.1<br>FCS_TLSC_EXT.2 | Syslog over TLS |
| ECC (P-384) | FCS_TLSC_EXT.1<br>FCS_TLSC_EXT.2 | Syslog over TLS<br>Distributed TOE Communication |
| ECC (P-256/384/521)<br>ECC (P-384) | FCS_TLSS_EXT.1 | HTTPS/TLS Remote Administration<br>Distributed TOE Communication |
| ECC (P-384) | FCS_TLSS_EXT.2 | Distributed TOE Communication |
| ECC (DH-19/20/21) | FCS_IPSEC_EXT.1.11 | IKEv2 key exchange |
| FFC safe primes (DH-14/15/16/17/18) | FCS_IPSEC_EXT.1.11 | IKEv2 key exchange |

- NDcPP30e:FCS_CKM.4: The TOE components clear keys (TLS and IPsec) from memory after those keys are no longer needed. After use on the SMC keys are overwritten with zeros and garbage collector is called. This is performed by the SMC Java Code and Forcepoint Network Security Platform SMC FIPS Java API 2.0.1 library. After use on the Security Engine, keys are overwritten with zeros. This is performed by the Forcepoint Network Security Platform FIPS Cryptographic Module.

  The TOE uses file system calls to clear persistently stored keys. On the SMC, TLS and IPsec private keys are stored in a Java Keystore. To clear these keys the disk must be wiped. This can be done via the installer by selecting the "Secure wipe with Automatic install". Data is sourced from /dev/random and then written to the disk. This is done 3 times for the whole disk.

  On the Security Engine TLS and IPsec private keys are stored in a flat file. To clear these keys the disk must be wiped. This can be done by resetting to factory defaults and choosing a number of overwrites. Data is sourced from /dev/random and then written to disk.

- NDcPP30e/VPNGW13:FCS_COP.1/DataEncryption: For TLS, the TOE performs encryption and decryption using AES in either CBC or GCM mode, and key sizes of either 128 or 256. For IPsec, the TOE uses AES in either CBC or GCM mode with key sizes of 128, 256 or 192. The crypto modules providing the AES implementation and the corresponding CAVP certificates are identified in **Table 4**, **Table 5**, **Table 6**, **Table 7**, and **Table 8**.

- NDcPP30e:FCS_COP.1/Hash: 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.

- NDcPP30e:FCS_COP.1/KeyedHash: The TOE supports keyed-hash message authentication using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 using SHA-1/256/384/512 with 160/256/384/512-bit keys to produce a 160/256/384/512 output MAC. The HMAC configuration is automatic based on other configuration entries. No administrator action is needed. The SHA-1/256 and 384/512 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).
  - IKE ESP supports all HMAC with SHA-1 (truncated to 96-bits as per RFC 2404) and SHA2 (256, 384, 512, not truncated) hashes for SA
  - TOE integrity check (the Virtual SMC Appliance checks its file system integrity using an ECDA digital signature verification of a catalog file of SHA-256 hashes of TOE binaries), and TLS 1.2 will use HMAC-SHA-1, HMAC-SHA-256 and HMAC-SHA-384 with a 160/256/384 bit key, respectively.

- NDcPP30e:FCS_COP.1/SigGen: The TOE supports the use of RSA with 2048, 3072 and 4096 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.

- NDcPP30e:FCS_HTTPS_EXT.1: The TOE provides an HTTPS/TLS interface for SMC Client GUI administration. The SMC Client GUI does not require or support client certificate authentication. The TOE implements the HTTPS protocol in accordance with RFC 2818.

- NDcPP30e/VPNGW13:FCS_IPSEC_EXT.1: The TOE processes (in conjunction with its firewall rules) both incoming and outgoing packets according to its administrator configured Security Policy Descriptors (SPDs) in order to apply (or not apply) IPsec processing. The TOE provides both site-to-site as well as

client IPsec encryption. The administrator can configure, for specific traffic, whether the TOE will discard (or drop), protect (encrypt with ESP), or bypass (forward without encryption) packets destined for specific peers, in compliance with RFC 4301. The administrator can configure the priority of the IPsec policies, and thus determine to order in which the TOE inspects (and applies) IPsec encryption policies (with the TOE applying the first matching policy) and the TOE recognizes packets from an established SA and automatically applies the encryption (or decryption) belonging to that SA.

The TOE supports tunnel mode alone, supports AES-CBC (128/192/256-bit keys) and AES-GCM (256 bit keys) for IKEv2, supports AES-CBC and GCM (128/192/256-bit keys) for ESP, supports all HMAC with SHA-1 (truncated to 96-bits as per RFC 2404) and SHA2 (256, 384, 512, not truncated) hashes for SA integrity, supports only IKEv2, supports administrator configuration of the maximum lifetime for IKEv2 and Child/ESP SAs, and also supports a maximum number of bytes for Child/ESP SAs. Administrator configuration of the maximum lifetime for IKEv2 and Child/ESP SAs and the maximum number of bytes for Child/ESP SAs is done when defining the properties of a VPN Profile.

The TOE supports DH groups 14-21 and, using its DRBG, generates a secret key ("x") that has a length of twice the security strength of the negotiated DH group (thus because the TOE supports DH groups 14-21, the secret key length ranges between 224 bits and 256 bits). The administrator can specify the DH Group in the IKEv2 policy. As a responder the TOE selects the first configured DH group proposed by the peer. Additionally, the TOE generates nonces during IKE key exchange with a minimum size of either 128-bits or half the size of the negotiated PRF hash (whichever is less).

The TOE prevents an administrator from misconfiguring the IKEv2 and Child/ESP SA cipher key lengths and ensures that the Child ESP SA's AES key length exceeds or equals that of the IKEv2 SA.

The TOE supports both RSA and ECDSA certificates for IKE peer authentication and can use SAN fields (IP, FQDN, or user FQDN) in addition to a peer certificate's Distinguished Name as a reference identifier.

The IKE SA uses IKEv2 with an available Pre-Shared Key authentication method. The IKE SA also has the option to configure the IKEv2 PPK as enabled or mandatory to use with IKEv2 protocol for the tunnel to be established. A PPK consists of a Key ID and a Pre-Shared secret. The secret can be one of the following: an ASCII string, a hex-encoded string if it has a 0x prefix in its value, or a Base64-encoded string if it has a 0s prefix in its value. The secret must have a minimum of 256 bits of entropy. The IPsec Client can be configured to allow Pre-Shared Key Authentication with IKEv1, which should be done when a third-party VPN client uses a pre-shared key for authenticating the VPN clients and the gateway. The pre-shared key is defined in the properties of User elements that have Pre-Shared Key Method as an authentication method. The User element defines who a user is and how they can identify themselves to get access to networks and services.

- NDcPP30e:FCS_NTP_EXT.1: The TOE provides the ability to synchronize its time with a NTP server using NTPv4. The time data is protected by a SHA1 message digest.
- NDcPP30e:FCS_RBG_EXT.1: The TOE components perform random bit generation in support of the cryptographic functions. The SMC uses a SHA-512 Hash_DRBG while the Engines use AES-256 CTR_DRBGs. The TOE components all use a software-based and a platform-based noise source; however, please note NIST SP 800-90B considers the platform-based noise source a secondary source that provides no entropy. Both the SMC and Engines draw from both sources when instantiating their respective DRBGs, and the software noise source supplies 512 bits of entropy to the Forcepoint Network Security Platform SMC FIPS Library 3.0.7 (based on Oracle Linux 9 OpenSSL FIPS Provider 3.0.7-b27cdeb3ba51be46)'s DRBG, supplies 512 bits of entropy to the Forcepoint Network Security Platform Security Engine FIPS Cryptographic Module 2.0 (based upon SafeZone FIPS Cryptographic Module 2.0) 's AES-256 CTR_DRBG, and supplies 384-bits of entropy to the Forcepoint Network Security Platform Security Engine FIPS Library 3.1.2 (based upon OpenSSL 3.1.2 FIPS)'s AES-256 CTR_DRBG.
- NDcPP30e:FCS_TLSC_EXT.1/2: The TOE communicates with both remote audit servers and other distributed TOE components using the TLS protocol. Mutual authentication using client-side x.509v3 certificates is supported by the SMC TLS client for syslog over TLS and for the TLS communication between the distributed TOE components. For TLS communication with remote audit servers, the administrator can configure a reference identifier of DNS name. 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. For communication with distributed TOE components, the TOE mandates the SAN:DNS extension. This expected DNS name is negotiated at the time of registration of the Security Engine to the SMC. 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 supports wildcards for TLS communication with a remote audit server. The TOE does not support wildcards and will always reject them for Distributed TOE communication.

The TOE presents the signature_algorithms extension with following supported signature algorithms:

**Syslog over TLS (SMC as Client)**
rsa_pkcs1 with sha256(0x0401),
rsa_pkcs1 with sha384(0x0501),
rsa_pkcs1 with sha512(0x0601),
ecdsa_secp256r1 with sha256(0x0403),
ecdsa_secp384r1 with sha384(0x0503),
ecdsa_secp521r1 with sha512(0x0603),
rsa_pss_rsae with sha256(0x0804),
rsa_pss_rsae with sha384(0x0805),
rsa_pss_rsae with sha512(0x0806),
rsa_pss_pss with sha256(0x0809),
rsa_pss_pss with sha384(0x080a),
rsa_pss_pss with sha512(0x080b)

**FPT_ITT (Engine as Client)**
ecdsa_secp384r1 with sha384(0x0503)

**FPT_ITT (SMC as Client)**
ecdsa_secp384r1 with sha384(0x0503)

The signature algorithms used for Syslog over TLS (SMC as Client) are configurable via the Web UI. The signature algorithms used for FPT_ITT (Engine as Client & SMC as Client) are limited to ecdsa_secp384r1 with sha384 (0x0503) because the evaluated configuration has FIPS mode enabled with the management server, log server, and security engines enrolled with an ECDSA key size of 384 bits (NIST curve P-384) which is not configurable.

- NDcPP30e:FCS_TLSS_EXT.1/2: The TOE provides a TLS interface for WebGUI 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 (WebUI only), P-384, or P-521 (WebUI only) curves based upon what the peer/client specifies it supports in its hello. The Web UI supports P-256, P-384, or P-521 for its certificates. The internal TOE communications support P-384 curves. 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. The administrator can, however, configure the list of ciphersuites that the TOE's TLS WebGUI will accept (using the "Server TLS Cryptography Suite Set" selection from the "Web Access" tab of the Management Server Properties dialog.

  The TOE also provides a mutually authenticated TLS server interface on the SMC and Engines to allow for secure, distributed TOE communications between those two components. Aside from the differences in authentication (mutually authenticated using Internal CA certificates), the internal TOE communications channels always uses TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 and needs no configuration outside FIPS mode. When the components exchange certificates as part of distributed TOE TLS handshake authentication, each side compares the received SAN:DNS to ensure that it matches the expected identifier of the component (the TOE's components require the presence of the SAN:DNS and do

not rely upon CN). The Engine does not support session resumption or session tickets. The SMC UI and SMC Server channels support session resumption based on session IDs according to RFC 5246. The SMC distributed TOE channel supports secure renegotiation, while the SMC Web UI and Engine distributed TOE channel reject renegotiation attempts. The Distributed TOE channels support the signature_algorithms extension, do not support the signature_algorithms_cert extension, and the behavior is not configurable.

## 6.4 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 satisfies the following security functional requirements:

- STFFW14e: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.

## 6.5 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 Security 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, ICMPv6, connections over IPv4 and IPv6. The Security Engine inspects and filters these protocols based upon their header fields as defined by their corresponding RFC (See Table 14).

The Security 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[3] | 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 14 Protocols & Fields Filtered by the TOE**

Any network traffic passed by the Security 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 no packets are ever passed that would violate policy).

All received network packets are processed by the Security Engine software module before transmission. The Forcepoint Network Security Platform 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 Security 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 Security Engine software module has been loaded, and the Security Engine software module denies all traffic until

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

the module has been configured. Network interfaces and routing are configured after the Security 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 Security Engine has been designed to ensure that no residual information exists in network packets. When the Security 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 Security 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.

| 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 15 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 Security 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 Security Engine applies the target actions. Possible target actions include Allow, Discard and Refuse[4]. 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 Security Engine compares the information attributes defined in Table 14 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 16 Additional Stateful Filtering Rules are applied.

| Rules relating to FFW_RUL_EXT.1.6 |
|---|

[4] Additional target actions of "Continue" and "Jump" described later support complex security policies.

| |
|---|
| a) The Security Engine denies and allows logging packets which are invalid fragments;<br>b) The Security Engine denies and allows logging fragmented packets which cannot be re-assembled completely;<br>c) The Security Engine denies and allows logging packets where the source address of the network packet is defined as being on a broadcast network;<br>d) The Security Engine denies and allows logging packets where the source address of the network packet is defined as being on a multicast network;<br>The Security Engine denies and allows logging network packets where the source address of the network packet is defined as being a loopback address;<br>e) The Security 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;<br>f) The Security 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 range: 2000::/3) as specified in RFC 3513 for IPv6;<br>g) The Security Engine denies and allows logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified;<br>Rules relating to FFW_RULE_EXT.1.7<br>a) The Security Engine denies and logs packets where the source address is equal to the address of the network interface where the network packet was received;<br>b) The Security Engine denies and logs packets where the source or destination address of the network packet is a linklocal address;<br>c) The Security Engine denies and logs 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 16 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 Security Engine obtains time values from the local hardware clock when making the security policy decisions associated with time-based information flows.

During the Security 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 TOE supports all IPv4 and IPv6 protocols except IPv6 protocol 1, which the TOE blocks by default as IP protocol 1 is ICMP for **IPv4** (while protocol 58 is IPv6 ICMP) and is not a valid protocol for IPv6.

The Firewall function satisfies the following security functional requirements:

- STFFW14e:FFW_RUL_EXT.1: The Security 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 Security 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.
- STFFW14e:FFW_RUL_EXT.2: The TOE performs stateful packet filtering on the FTP protocol. No protocols cause automatic generation of dynamic packet filtering rules, rather, the TOE implements connection tracking as described above.

## 6.6 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 of between 1 and 80 characters for passwords. By default, the minimum password length is 10 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 Security Engine prior to an administrative user being authenticated. The TOE requires login prior to allowing any TOE configuration actions.

The Identification and authentication function satisfies the following security functional requirements:

- NDcPP30e:FIA_AFL.1: The TOE allows the administrator to specify the maximum number of incorrect logins as well as lock out period for an administrator who exceeds the maximum configured value. The administrator can set the number of failed attempts to a value from 1-1000 and the lockout duration from 1-1000 (and choose from minutes, hours, days). The TOE defaults to 6 incorrect attempts and a 30 minute lock out period. The local CLI remains available when the remote account is locked out.
- NDcPP30e:FIA_PMG_EXT.1: Password for local accounts can be composed of upper or lower case letters, numbers, and special characters as described above. The minimum password length is configurable from 1 to 80 characters. By default, the TOE enforces a minimum password length of 10 characters. When operating in a Common Criteria evaluated configuration, the recommended minimum password length is 15 characters.
- VPNGW13:FIA_PSK_EXT.1/2: The TOE supports externally generated bit-based postquantum pre-shared keys (PPK) that conform to RFC 8784 for IKE authentication. When using PPKs the TOE should be configured for PPKs to be mandatory.
- NDcPP30e:FIA_UAU.7: All passwords entered by administrators are obscured when entered.
- NDcPP30e:FIA_UIA_EXT.1: Prior to administrative login, the TOE displays a banner, and filters network traffic. The TOE requires login prior to all administrative actions. The SMC Management Server only accepts TLSv1.2 connections for management operations. When a valid username/password is provided by the administrator the TOE logs in the administrator, giving them access to the GUI or CLI commands.
- NDcPP30e/VPNGW13:FIA_X509_EXT.1/ITT/Rev: The TOE supports OCSP and CRL revocations for X509v3 certificate validation during negotiation of TLS protected syslog and during IKEv2 authentication. 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). The TOE performs no revocation checking as part of distributed TOE TLS communications.

- NDcPP30e/VPNGW13:FIA_X509_EXT.2: The administrator explicitly configures the certificate to be used by the TOE for each individual channel on both the SMC and engine thus the TOE knows exactly what certificate to present for authentication. The TOE performs revocation checking when validating a server certificate during TLS establishment with a remote syslog server or when authenticating an IKEv2 peer's certificate. The TOE performs no revocation checking as part of distributed TOE TLS communications. 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 not 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.
- NDcPP30e:FIA_X509_EXT.3: The TOE generates certificate requests and validates the CA used to sign the certificate. The TOE generates certificate requests which include public key, Common Name, Organization, Organizational Unit, Country and device-specific information in the form of Subject Alternative Name. Administrators can generate the CSR via the SMC GUI and then the CSR is manually sent to a Certificate Authority (CA) for the CA to sign and issue a certificate. Once the certificate has been issued, the administrator can import the X.509v3 certificate into the TOE along with the CA certificate(s) that signed and issued the server (syslog) certificate. This allows the TOE to determine which CA certificate(s) to use during the validation process. If the TOE does not find the trusted root CA, the trusted channel establishment (TLS connections to the syslog server or IKEv2/IPsec connections) will fail. For distributed TOE communication, the TOE uses internal certificates that are automatically generated during installation and registration. When the SMC is installed, an internal ECDSA certificate authority is automatically created and issues certificates for the SMC. The Security Engine certificates are issued during the registration process when the SMC sends the internal CA certificate to the Security Engine. For further details regarding how to configure the TOE to use certificates, please refer to the Common Criteria Evaluated Configuration Guide.

## 6.7 Security management

The TOE provides an administrator role. User accounts that are associated with the administrator role are considered Security Administrators. Users in this role can perform security management functions locally and remotely. Security Administrators can manage and configure all TSF data including audit data, cryptographic data, authentication data, configuration data, user and administrator security attributes, session timeouts, and updates.

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

The Virtual SMC Appliance 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 the commands available only through the CLI for manually setting the time and configuring the CLI session time out.

The Security management function satisfies the following security functional requirements:.

- NDcPP30e:FMT_MOF.1/Functions: The TOE allows only authenticated administrators to configure and view the configuration of TLS protected syslog export and to determine the handling of audit data if local audit storage space becomes full by viewing the alert. The administrator can resume the collection of audit data, by deleting the logs to reduces audit storage. This administration is only performed from the SMC.

- NDcPP30e:FMT_MOF.1/ManualUpdate: Only the administrator can initiate product updates. The administrator initiates the updates of both the SMC and the Engines through the SMC. Each type of component performs the required cryptographic signature checks when updating.
- NDcPP30e:FMT_MTD.1/CoreData: The TOE ensures that only security administrators can login to manage TSF data and configure TOE services. TSF data includes audit data, cryptographic data, authentication data, configuration data, security attributes, session timeouts, and updates. The TOE requires that the administrator perform all configuration through the SMC (which then communicates with the Engines under its control)—the Engines provide no direct interface for administrators in the evaluated configuration.
- NDcPP30e/VPNGW13:FMT_MTD.1/CryptoKeys: The TOE allows only authenticated administrators to configure (import, generate, delete, change) cryptographic keys.
- NDcPP30e/VPNGW13:FMT_SMF.1: Administrators can configure operations of the TOE through the GUI, including configuring cryptographic functionality, audit behavior, authentication failure parameters, cryptographic keys, the reference identifier for peers (external syslog server and IKEv2 peers), services available prior to login, remote session inactivity timeout, and configuring NTP. The local command line interface is used by administrators to query the current SMC firmware version, install SMC updates, manually set the time, and configure the CLI session timeout. The administrator can enable the interaction between TOE components (the TOE only allows communications between the SMC and each of the Engines under its control) as part of the setup process. The administrator can disable the interaction between TOE components by removing the Engine from the SMC's control. Administrators can import X.509v3 certificates to the TOE's trust store and designate X509.v3 certificates as trusted certificate authorities. The administrator can define packet filtering rules, associate those rules with an interface and define priority to the rules by ordering. The administrator can also define the timeout for VPN client sessions and the attributes for denying VPN client sessions.
- STFFW14e:FMT_SMF.1/FFW: Administrators can configure the TOE firewall rules and policies using the GUI.
- NDcPP30e:FMT_SMR.2: The TOE maintains an administrative role for users. Users in this role can perform administrative actions locally or remotely.

## 6.8 Packet Filtering

The packet filtering function is the VPN extended package is addressed entirely by the FFW_RUL_EXT.1 requirement. See Section 6.5.

The Packet Filtering function satisfies the following security functional requirements:

- VPNGW13:FPF_RUL_EXT.1: Please see Section 6.5 Firewall above for a description of the TOE's packet filtering capabilities.

## 6.9 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 distributed TOE communication between the Security Engine and the servers running on the Virtual SMC Appliance are all protected using TLS network connections.

The keys retained by the components of the TOE are associated with certificates used for TLS. The servers running on the Virtual SMC Appliance store private keys in a password protected Java keystore.

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 through console administration of the SMC Management Server or by enabling NTP time synchronization via the SMC GUI. Time on the Security Engine is updated by the SMC Management Server or alternatively from an administrator configured NTP server.

Each component of the TOE includes a set of hardware validation tests which include noise source health tests (the NIST SP 800-90B approved continuous health tests), and Known Answer Tests (KAT) for the cryptographic features provided by the Forcepoint Network Security Platform Security Engine FIPS Library 3.1.2 (based upon OpenSSL 3.1.2 FIPS), Virtual SMC Appliance Forcepoint Network Security Platform SMC FIPS Library 3.0.7 (based on Oracle Linux 9 OpenSSL FIPS Provider 3.0.7-b27cdeb3ba51be46), Virtual SMC Appliance Forcepoint Network Security Platform SMC FIPS Java API 2.0.1, and Forcepoint Network Security Platform SMC FIPS Cryptographic Module for NTP 3.101 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 Security Engine will reboot as a result of the error, while the Virtual SMC Appliance will halt its boot as a result of a KAT error or any error in its integrity verification (the SMC verifies the ECDSA P-521 with SHA2-384 signature on a catalog file of SHA-256 digests of the SMC binaries) upon system startup. The Security Engine (using its Forcepoint Network Security Platform Security Engine FIPS Library 3.1.2 (based upon OpenSSL 3.1.2 FIPS)) uses a preinstalled public key to verify the ECDSA P-521 with SHA-512 signature of the whole partition containing TOE binaries (and only TOE binaries) and if it finds an error, it will reboot. These integrity verification keys are included in the Forcepoint software and the TOE provides administrators no method to access them. These tests are sufficient because any TOE modifications or failed cryptographic operations will be immediately noted upon boot.

The TOE performs trusted updates for both of its components: the Virtual SMC management appliance and Security Engines. To update the TOE software of the Security 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 Security Engines with the new software. If the signature verification fails, the update package will not be available for install. Once the administrator selects to upgrade a specific Security Engine with a patch, the SMC will transfer that update to the Security Engine. The Engine will use that update package (which is a 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. The administrator can make the patch available to the SMC two different ways, either by saving the patch to an administrator provided USB thumb-drive which is then mounted to the SMC or by uploading it to the SMC through the GUI. 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, the administrator can issue the ambr_install command to install the patch, and then follow the installation process (which can require a reboot for upgrades or major new features). Note the administrator can also execute the ambr_install command and it will perform the signature verification automatically and only install the patch if the signature check succeeds.

The Protection of the TSF function satisfies the following security functional requirements:

- NDcPP30e:FPT_APW_EXT.1: Passwords are stored on the Virtual 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.
- VPNGW13:FPT_FLS.1/SelfTest: The TOE will halt boot (or reboot) should any of its self-tests fail (power-up self-tests, executable image integrity check, noise source health tests.
- NDcPP30e:FPT_ITT.1: The TOE utilizes TLS protected communications from Engines to their SMC (log forwarding) and from the SMC to its Engines (management). The TOE has no other communications between components.

- NDcPP30e: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 Virtual SMC Appliance) and there exists no interface to view them. The keys exist on a Read/Write (RW) partition on the Security Engine. Since the Security Engine does not support an interface for local administration, this data is not accessible once stored in the partition..
- NDcPP30e:FPT_STM_EXT.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 through console administration of the SMC Management Server or by enabling NTP time synchronization via the SMC GUI. Time on the Security Engine is updated by the SMC Management Server or alternatively from an administrator configured NTP server.
- NDcPP30e/VPNGW13:FPT_TST_EXT.1/VPNGW13:FPT_TST_EXT.3: The TOE components verify cryptographic algorithms and checksums of TOE binaries upon startup as described above.
- NDcPP30e/VPNGW13:FPT_TUD_EXT.1: The administrator can query the current software versions for the SMC software on the SMC component and for the Security Engine software on the SMC Management Server as well as the most recently installed version of the TOE software. Administrators can obtain TOE patches from Forcepoint. Administrators must initiate the installation of patches to the Security Engine and to the Virtual 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. If the administrator wishes to revert to the previous software version, the device can be rebooted and the previous version selected as the boot option.

## 6.10 TOE access

The GUI offered by the Virtual 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 Virtual 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 satisfies the following security functional requirements:

- NDcPP30e: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.
- VPNGW13:FTA_SSL.3/VPN: The TOE can also terminate VPN client sessions that have been inactive for the administrator defined time interval.
- NDcPP30e: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.
- NDcPP30e:FTA_SSL_EXT.1: The only local interactive sessions are those offered by the Virtual SMC Appliance providing a command line interface.
- NDcPP30e:FTA_TAB.1: A Banner is displayed on both interfaces offered by the Virtual SMC Appliance (i.e., the GUI and local console). The Security Engine does not offer a direct network interface.
- VPNGW13:FTA_TSE.1: The TOE can deny VPN client sessions based upon location (as determined by IP address), time and day.
- VPNGW13:FTA_VCM_EXT.1: An administrator can configure the TOE to assign private IP addresses to VPN clients.

## 6.11 Trusted path/channels

The TOE has trusted external IT entity communications with an external syslog server (protected by TLS) and with VPN peers (protected by IPsec).

For the syslog communication channel, the TOE acts as the TLS client during the negotiation of the TLS connection. 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.

For communications with VPN Clients, the TOE acts as an IKE responder, while for communications with VPN peers, the TOE can act as an IKE initiator or a responder. In all cases, the TOE authenticates IKE peers and Clients through certificate exchange.

The administrator's Client GUI runs within an Internet browser running on the administrator's workstation and provides 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 TLSv1.2 connections for the Client GUI and provides an HTTPS/TLS interface for using the Client GUI in a web browser. An administrator can also configure the TOE to provide VPN Clients with access to the SMC's Management Server, allowing a remote administrator to securely access the SMC through a browser tunneled within IPsec.

The Trusted path/channels function satisfies the following security functional requirements:

- NDcPP30e:FTP_ITC.1: The TOE protects syslog communication from the Virtual SMC Appliance's Log Server and Management Server to the external syslog server using the TLSv1.2 protocol.
- VPNGW13:FTP_ITC.1/VPN: The TOE's Engines can also protect traffic with remote IPsec peers (either site-to-site or VPN clients).
- NDcPP30e:FTP_TRP.1/Admin: The SMC Management Server's Web UI accepts only TLSv1.2 connections for management operations. The TOE can be configured to provide VPN Clients with access to the SMC's Management Server, allowing a remote administrator to securely access the SMC through a browser tunneled within IPsec.
- NDcPP30e:FTP_TRP.1/Join: The SMC Management Server only accepts join requests from Engines for which the Administrator has already created an object and provided the Engine with the SMC generated password. Furthermore, the SMC protects the registration channel using TLSv1.2 and requires that the registering Engine prove knowledge of the SMC generated password by exchanging a SHA-512 hash. The SMC reserves port 3021 for registration and after registration, the components communicate using mutually-authenticated TLS on different ports (8903, 8907, 8906, 8916, 8917, 3020, 3023), thus preventing reuse of the registration channel. Should a registration attempt fail, the administrator can attempt again, or can generate a new password on the SMC (and input that password into the Engine), and then attempt registration again.

# 7. Requirement Allocation

This section provides a mapping of the distributed TOE components to the SFRs in this ST. This TOE is a distributed TOE consistent with Use Case 3 as defined in the NDcPP30e. The following table presents the required mapping.

| Requirement | Distributed TOE SFR Allocation | Distributed TOE Audit Event Allocation |
|---|---|---|
| **NDcPP30e/STFFW14e:FAU_GEN.1** | All | All |
| **NDcPP30e:FAU_GEN.2** | All | None |
| **VPNGW13:FAU_GEN.1/VPN** | Engine(s) | Engine(s) |
| **NDcPP30e:FAU_GEN_EXT.1** | All | All |
| **NDcPP30e:FAU_STG_EXT.1** | All | None |
| **NDcPP30e:FAU_STG_EXT.4** | All | None |
| **NDcPP30e:FAU_STG_EXT.5** | All | None |
| **NDcPP30e:FCO_CPC_EXT.1** | All | All |
| **NDcPP30e:FCS_CKM.1** | All | None |
| **VPNGW13:FCS_CKM.1/IKE** | Engine(s) | Engine(s) |
| **NDcPP30e:FCS_CKM.2** | All | None |
| **NDcPP30e:FCS_CKM.4** | All | None |
| **NDcPP30e:FCS_COP.1/DataEncryption** | All | None |
| **VPNGW13:FCS_COP.1/DataEncryption** | Engine(s) | Engine(s) |
| **NDcPP30e:FCS_COP.1/SigGen** | All | None |
| **NDcPP30e:FCS_COP.1/Hash** | All | None |
| **NDcPP30e:FCS_COP.1/KeyedHash** | All | None |
| **NDcPP30e:FCS_HTTPS_EXT.1** | SMC | SMC |
| **NDcPP30e:FCS_NTP_EXT.1** | All | All |
| **NDcPP30e:FCS_RBG_EXT.1** | All | None |
| **NDcPP30e:FCS_TLSC_EXT.1** | All | All |
| **NDcPP30e:FCS_TLSC_EXT.2** | All | All |
| **NDcPP30e:FCS_TLSS_EXT.1** | All | All |
| **NDcPP30e:FCS_TLSS_EXT.2** | All | All |
| **NDcPP30e:FCS_IPSEC_EXT.1** | Engine(s) | Engine(s) |
| **VPNGW13:FCS_IPSEC_EXT.1** | Engine(s) | Engine(s) |
| **STFFW14e:FDP_RIP.2** | All | None |
| **STFFW14e:FFW_RUL_EXT.1** | Engine(s) | Engine(s) |
| **STFFW14e:FFW_RUL_EXT.2** | Engine(s) | None |
| **NDcPP30e:FIA_AFL.1** | SMC | SMC |
| **NDcPP30e:FIA_PMG_EXT.1** | SMC | None |
| **VPNGW13:FIA_PSK_EXT.1** | Engine(s) | Engine(s) |
| **VPNGW13:FIA_PSK_EXT.2** | Engine(s) | Engine(s) |
| **NDcPP30e:FIA_UAU.7** | SMC | None |
| **NDcPP30e:FIA_UIA_EXT.1** | SMC | SMC |
| **NDcPP30e:FIA_X509_EXT.1/ITT** | All | All |
| **NDcPP30e:FIA_X509_EXT.1/Rev** | All | All |
| **VPNGW13:FIA_X509_EXT.1/Rev** | Engine(s) | Engine(s) |
| **NDcPP30e:FIA_X509_EXT.2** | All | None |
| **VPNGW13:FIA_X509_EXT.2** | Engine(s) | Engine(s) |
| **NDcPP30e:FIA_X509_EXT.3** | SMC | None |
| **VPNGW13:FIA_X509_EXT.3** | Engine(s) | Engine(s) |
| **NDcPP30e:FMT_MOF.1/Functions** | SMC | SMC |

| Requirement | Distributed TOE SFR Allocation | Distributed TOE Audit Event Allocation |
|---|---|---|
| **NDcPP30e:FMT_MOF.1/ManualUpdate** | All | All |
| **NDcPP30e:FMT_MTD.1/CoreData** | All | SMC |
| **NDcPP30e:FMT_ MTD.1/CryptoKeys** | All | All |
| **VPNGW13:FMT_ MTD.1/CryptoKeys** | All | SMC |
| **NDcPP30e:FMT_SMF.1** | SMC | None |
| **STFFW14e:FMT_SMF.1/FFW** | SMC | None |
| **VPNGW13:FMT_SMF.1/VPN** | SMC | None |
| **NDcPP30e:FMT_SMR.2** | SMC | None |
| **VPNGW13:FPF_RUL_EXT.1** | SMC | None |
| **NDcPP30e:FPT_APW_EXT.1** | SMC | None |
| **VPNGW13:FPT_FLS.1/SelfTest** | All | All |
| **NDcPP30e:FPT_ITT.1** | All | All |
| **NDcPP30e:FPT_SKP_EXT.1** | All | None |
| **NDcPP30e:FPT_STM_EXT.1** | All | All |
| **NDcPP30e:FPT_TST_EXT.1** | All | None |
| **VPNGW13:FPT_TST_EXT.1** | All | None |
| **VPNGW13:FPT_TST_EXT.3** | All | None |
| **NDcPP30e:FPT_TUD_EXT.1** | All | All |
| **VPNGW13:FPT_TUD_EXT.1** | All | All |
| **NDcPP30e:FTA_SSL.3** | SMC | SMC |
| **VPNGW13:FTA_SSL.3/VPN** | SMC | SMC |
| **NDcPP30e:FTA_SSL.4** | SMC | SMC |
| **NDcPP30e:FTA_SSL_EXT.1** | SMC | SMC |
| **NDcPP30e:FTA_TAB.1** | SMC | None |
| **VPNGW13:FTA_TSE.1** | Engine(s) | Engine(s) |
| **VPNGW13:FTA_VCM_EXT.1** | Engine(s) | Engine(s) |
| **NDcPP30e:FTP_ITC.1** | SMC | SMC |
| **VPNGW13:FTP_ITC.1/VPN** | Engine(s) | Engine(s) |
| **NDcPP30e:FTP_TRP.1/Admin** | SMC | SMC |
| **NDcPP30e:FTP_TRP.1/Join** | All | All |