Forcepoint NGFW 6.10.9 Security Target

Version 0.5
04/14/2023

Prepared for:

Forcepoint

10900-A Stonelake Blvd.
Austin, TX 78759, USA
www.forcepoint.com

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 Forcepoint NGFW provided by Forcepoint. 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 the ST, iteration is indicated by a parenthetical number placed at the end of the component. For example FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement.
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
  - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.1 Security Target Reference

ST Title – Forcepoint NGFW 6.10.9 Security Target

ST Version – Version 0.5

ST Date – 04/14/2023

1.2 TOE Reference

TOE Identification – Forcepoint NGFW 6.10.9

TOE Developer – Forcepoint

Evaluation Sponsor – Forcepoint

1.3 TOE Overview

The Target of Evaluation (TOE) is the Forcepoint NGFW 6.10.9.

The Forcepoint Next Generation Firewall is a stateful packet filtering firewall and VPN gateway. Being a stateful packet filtering firewall, the NGFW filters network traffic optimized through the use of stateful packet inspection. Being a VPN gateway, the NGFW provides IPsec VPN functionality to secure network data exchanged with peer gateways and VPN clients. The NGFW is intended to be used as a network perimeter security gateway that provides a controlled connection. The NGFW is centrally managed and generates audit records for security critical events.

1.4 TOE Description

The Forcepoint Next Generation Firewall is a stateful packet filtering firewall and VPN gateway. The Forcepoint Next Generation Firewall (NGFW) system is composed of the NGFW Engine (a physical or virtual appliance) and the Virtual Security Management Center (SMC). The NGFW 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 NGFW Engines. Throughout the remainder of this document, references to the NGFW Engine are meant to reference the TOE's firewall engine, while references to the NGFW are meant to refer to the TOE as a whole.

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

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

The NGFW Engine contains a hardened Linux operating system (with a 4.19 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 NGFW Engine, the SMC contains a hardened Linux-based operating system (which uses a 4.18 kernel) to support the management capabilities and allow for the operation and configuration of firewall engines.

1.4.1 TOE Architecture

The Forcepoint Next Generation Firewall (NGFW) system is a distributed TOE[1] consisting of the Security Management Center (SMC) Appliance and one or more NGFW Engines under the control of the SMC. These NGFW 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 upon both Java and native cryptographic libraries for cryptographic functionality. In the evaluated configuration, the Virtual SMC Appliance communicates with NGFW Engines through a TLS-protected trusted channel.

[1] The TOE is a distributed TOE consistent with Use Case 3 as defined in the NDcPP22e.

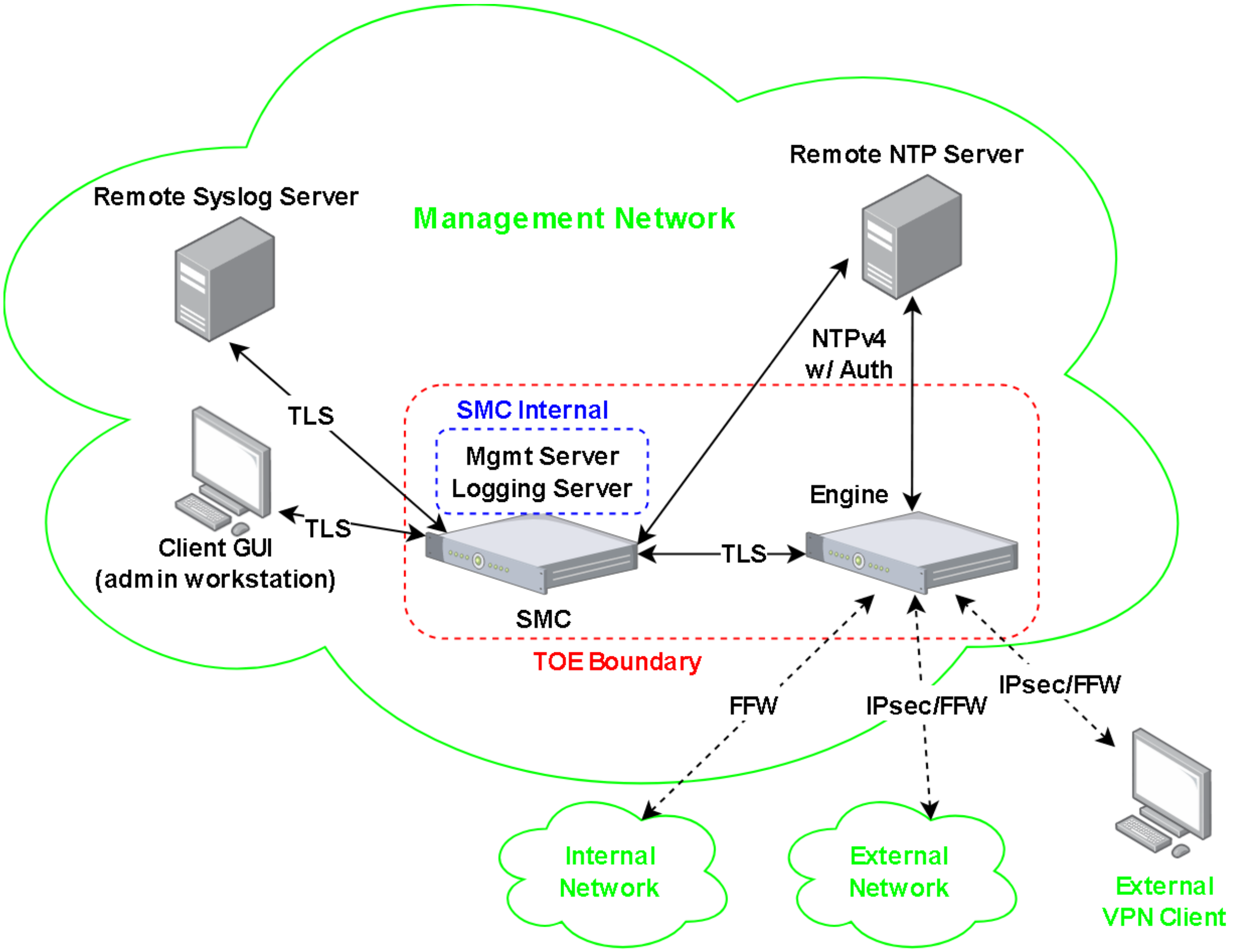

Figure 1 TOE Components, Communication Paths and IT Environment.

The NGFW Engines (a.k.a., the Engines) are responsible for performing all firewall packet handling, analysis and filtering that is provided by the NGFW 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 NGFW system through the SMC Client GUI. The Virtual SMC Appliance provides a very limited console interface that allows administrators to verify and update TOE software, to manually set the time, and configure the console timeout.

The NGFW 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 NGFW 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 NGFW Engine components of the TOE and securely re-transmitting the audit data to an external syslog server. The Management Server component directly transmits its audit data to an external syslog server.

The administrator interfaces with the TOE through a Management Client GUI (either the Forcepoint standalone Java Client installed from a Forcepoint provided installation package or through an HTML5 web browser application). The Client GUI (along with the administrator's workstation on which the Client is installed), 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 NGFW 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.
- **NGFW Engine to External NTP server communications** use SHA1 as the message digest algorithms for authentication with an NTP time source. Time on the NGFW Engine is updated by the SMC Management Server, or alternatively from an administrator configured NTP server.
- **NGFW Engine to Log Server communications** use the TLS-based trusted channel to protect the audit data transmitted from the NGFW Engine to the Log Server.
- **NGFW Engine to/from Management Server communications** use the TLS-based trusted channel to protect the configuration information exchanged between the Management Server and the NGFW Engine. Either party in this communication pathway can initiate the communications. Typically, the Management Server initiates configuration changes by sending updated security policies to the NGFW Engine. However, the NGFW Engine also polls for configuration changes on a regular basis.
- **Client GUI to Management and Log Server communications** uses TLS to protect the communication over which remote administration actions occur.
- The **NGFW Engines** control connectivity and information flow between **internal and external connected networks** that they are protecting.
- The **NGFW 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 SMC FIPS Java API 1.0.2.3 (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 OpenSSL library to perform signature verification supporting the TOE trusted update mechanism. The Virtual SMC Appliance's NTP daemon uses cryptography from the SMC FIPS Cryptographic Module for NTP 3.79.

The NGFW Engine utilizes its Forcepoint NGFW FIPS Library 1.1.1 to provide the encryption, decryption, signing and hashing functions necessary to support the NGFW Engine's trusted update mechanism and its TLS, ITT secure channel.

1.4.1.1 Physical Boundaries

The TOE is composed of one or more NGFW 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 NGFW Engines to monitor and filter network traffic. The Virtual SMC Appliance provides all management functionality, while the NGFW Engines provide all firewall packet filtering.

The TOE is accessed and managed from the Forcepoint Security Management Center Client (6.10.9) 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 NGFW Engine does not send audit data directly to an external syslog server. Instead, a NGFW Engine passes all of its audit data to the Log Server on the Virtual SMC Appliance, which can (if configured) forward the data to the external syslog server.

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 NGFW Engine's time to be in sync with itself. The NGFW Engine synchronizes only with the SMC, but can alternatively be configured separately to receive time from an NTP server directly.

The NGFW Engine utilizes its Forcepoint NGFW FIPS Library 1.1.1 based upon OpenSSL 1.1.1 (which utilizes the Forcepoint NGFW FIPS Cryptographic Module 1.2.1) to verify trusted engine software updates. The Virtual SMC Appliance uses its SMC FIPS Java API 1.0.2.3 to provide TLS (which protects the trusted channel mechanism and the trusted path mechanism) and uses its SMC FIPS Library 1.1.1 based on OpenSSL 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
N120	Desktop Intel Atom C3338(Denverton)	8	8	0	0
N120W	Desktop Intel Atom C3338(Denverton)	8	8	0	0
N120WL	Desktop Intel Atom C3338(Denverton)	8	8	0	0
N120L	Desktop Intel Atom C3338(Denverton)	8	8	0	0
N60	Desktop Intel Atom C3338(Denverton)	4	4	0	0
N60L	Desktop Intel Atom C3338(Denverton)	4	4	0	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
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
3410	2U Intel Xeon Gold 6230N (Cascade Lake)	1x GE RJ45, 2x 10Gbps SFP+	1 to 65	2 to 66	8
ESXi 7.0	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 7.0 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

- N60 Desktop Atom C3338 (Denverton)

1U Rack Mounted Firewall models

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

2U Rack Mounted Firewall models

- 3401 2U Xeon 4210 (Cascade Lake)

Virtual NGFW Engine Appliance

- VMWare ESXi 7.0 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 NGFW:

- 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 NGFW Engines. The Security Management Center can manage one or more NGFW 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, and the SMC uses cryptography to support its use of the TLS protocol 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 NGFW Engine enforces the firewall security policy on all traffic that passes through the engine, via its internal or external network Ethernet interfaces.

1.4.1.2.6 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 NGFW 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 performs self-tests that cover the correct operation of the TOE. It provides functions necessary to securely update the TOE. It's Linux-based operating system utilizes a hardware clock to ensure reliable timestamps. It protects sensitive data such as stored passwords and cryptographic keys so that they are not accessible through the TOE, even to a Security Administrator.

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.

1.4.1.2.11 Trusted path/channels

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

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

The TOE 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.

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.

1.4.2 TOE Documentation

The following administrator and user guidance is available:

- Forcepoint Next Generation Firewall Common Criteria Evaluated Configuration Guide, 6.10.9, Rev C

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 1.2, 31 March 2022
        - Base-PP: collaborative Protection Profile for Network Devices, Version 2.2e, 23 March 2020 (NDcPP22e)
        - PP-Module: PP-Module for Stateful Traffic Filter Firewalls, Version 1.4 + Errata 20200625, 25 June 2020 (STFFW14)
        - PP-Module: PP-Module for VPN Gateways, Version 1.2, 31 March 2022 (VPNGW12)

Package	Technical Decision	Applied	Notes
VPNGW12	TD0723 - Correction to ECDSA Curve Selection	Yes	
VPNGW12	TD0683 - RFC 2460 to be replaced with RFC 8200	Yes	
NDcPP22e	TD0670 - NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	Yes	
VPNGW12	TD0657 - IPSEC_EXT.1.6 GCM support for VPN GW	Yes	
VPNGW12	TD0656 - Missing EAs for VPN GW Optional Headend SFRs	Yes	
NDcPP22e	TD0639 - NIT Technical Decision for Clarification for NTP MAC Keys	Yes	
NDcPP22e	TD0638 - NIT Technical Decision for Key Pair Generation for Authentication	Yes	
NDcPP22e	TD0636 - NIT Technical Decision for Clarification of Public Key User Authentication for SSH	No	SSH not claimed
NDcPP22e	TD0635 - NIT Technical Decision for TLS Server and Key Agreement Parameters	Yes	
NDcPP22e	TD0634 - NIT Technical Decision for Clarification required for testing IPv6	Yes	TLSC claimed
NDcPP22e	TD0633 - NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	Yes	
NDcPP22e	TD0632 - NIT Technical Decision for Consistency with Time Data for vNDs	Yes	
NDcPP22e	TD0631 - NIT Technical Decision for Clarification of public key authentication for SSH Server	No	SSH not claimed
NDcPP22e	TD0592 - NIT Technical Decision for Local Storage of Audit Records	Yes	
NDcPP22e	TD0591 - NIT Technical Decision for Virtual TOEs and hypervisors	Yes	
NDcPP22e	TD0581 - NIT Technical Decision for Elliptic curve-based key	Yes	

	establishment and NIST SP 800-56Arev3		
NDcPP22e	TD0580 - NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	Yes	
NDcPP22e	TD0572 - NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	Yes	
NDcPP22e	TD0571 - NiT Technical Decision for Guidance on how to handle FIA_AFL.1	Yes	
NDcPP22e	TD0570 - NiT Technical Decision for Clarification about FIA_AFL.1	Yes	
NDcPP22e	TD0569 - NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	Yes	
NDcPP22e	TD0564 - NiT Technical Decision for Vulnerability Analysis Search Criteria	Yes	
NDcPP22e	TD0563 - NiT Technical Decision for Clarification of audit date information	Yes	
NDcPP22e	TD0556 - NIT Technical Decision for RFC 5077 question	Yes	
NDcPP22e	TD0555 - NIT Technical Decision for RFC Reference incorrect in TLSS Test	Yes	
NDcPP22e	TD0547 - NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	Yes	
NDcPP22e	TD0546 - NIT Technical Decision for DTLS - clarification of Application Note 63	No	FCS_DTLSC_EXT.1 not claimed
NDcPP22e	TD0538 - NIT Technical Decision for Outdated link to allowed-with list	Yes	
NDcPP22e	TD0537 - NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	Yes	
NDcPP22e	TD0536 - NIT Technical Decision for Update Verification Inconsistency	Yes	
NDcPP22e	TD0528 - NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	Yes	
NDcPP22e	TD0527 - Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	Yes	
STFFW14	TD0551 - NIT Technical Decision for Incomplete Mappings of OEs in FW Module v1.4+Errata	Yes	
STFFW14	TD0545 - NIT Technical Decision for Conflicting FW rules cannot be configured (extension of RfI#201837)	Yes	

2.1 Conformance Rationale

The ST conforms to the NDcPP22e/STFFW14e/VPNGW12. 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 NDcPP22e/STFFW14e/VPNGW12 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDcPP22e/STFFW14e/VPNGW12 offers additional information about the identified security objectives, but that has not been reproduced here and the NDcPP22e/STFFW14e/VPNGW12 should be consulted if there is interest in that material.

In general, the NDcPP22e/STFFW14e/VPNGW12 has defined Security Objectives appropriate for network device and as such are applicable to the Forcepoint NGFW 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 See TD0520 for SARs.
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 TOE 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 virtualisation 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 NDcPP22e/STFFW14e/VPNGW12. The NDcPP22e/STFFW14e/VPNGW12 defines the following extended requirements and since they are not redefined in this ST the NDcPP22e/STFFW14e/VPNGW12 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- NDcPP22e:FAU_GEN_EXT.1: Security Audit Generation
- NDcPP22e:FAU_STG_EXT.1: Protected Audit Event Storage
- NDcPP22e:FAU_STG_EXT.4: Protected Local Audit Event Storage for Distributed TOEs
- NDcPP22e:FAU_STG_EXT.5: Protected Remote Audit Event Storage for Distributed TOEs
- NDcPP22e:FCO_CPC_EXT.1: Component Registration Channel Definition
- NDcPP22e:FCS_HTTPS_EXT.1: HTTPS Protocol
- NDcPP22e:FCS_IPSEC_EXT.1: IPsec Protocol - per TD0633
- VPNGW12:FCS_IPSEC_EXT.1: IPsec Protocol - per TD0657
- NDcPP22e:FCS_NTP_EXT.1: NTP Protocol
- NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation
- NDcPP22e:FCS_TLSC_EXT.1: TLS Client Protocol Without Mutual Authentication - per TD0634
- NDcPP22e:FCS_TLSC_EXT.2: TLS Client Support for Mutual Authentication
- NDcPP22e:FCS_TLSS_EXT.1: TLS Server Protocol Without Mutual Authentication - per TD0635
- NDcPP22e: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
- NDcPP22e:FIA_PMG_EXT.1: Password Management
- NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism
- NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication
- NDcPP22e:FIA_X509_EXT.1/ITT: X.509 Certificate Validation
- NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- VPNGW12:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- NDcPP22e:FIA_X509_EXT.2: X.509 Certificate Authentication
- VPNGW12:FIA_X509_EXT.2: X.509 Certificate Authentication
- NDcPP22e:FIA_X509_EXT.3: X.509 Certificate Requests
- VPNGW12:FIA_X509_EXT.3: X.509 Certificate Requests
- VPNGW12:FPF_RUL_EXT.1: Packet Filtering Rules
- NDcPP22e:FPT_APW_EXT.1: Protection of Administrator Passwords
- NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
- NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps - per TD0632

- NDcPP22e:FPT_TST_EXT.1: TSF testing
- VPNGW12:FPT_TST_EXT.1: TSF Testing
- VPNGW12:FPT_TST_EXT.3: Self-Test with Defined Methods
- NDcPP22e:FPT_TUD_EXT.1: Trusted update
- VPNGW12:FPT_TUD_EXT.1: Trusted Update
- NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking
- VPNGW12:FTA_VCM_EXT.1: VPN Client Management - per TD0656

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 NDcPP22e/STFFW14e/VPNGW12. The refinements and operations already performed in the NDcPP22e/STFFW14e/VPNGW12 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDcPP22e/STFFW14e/VPNGW12 and any residual operations have been completed herein. Of particular note, the NDcPP22e/STFFW14e/VPNGW12 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 NDcPP22e/STFFW14e/VPNGW12. The NDcPP22e/STFFW14e/VPNGW12 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by Forcepoint NGFW TOE.

Requirement Class	Requirement Component
FAU: Security audit	NDcPP22e:FAU_GEN.1: Audit Data Generation
	STFFW14e:FAU_GEN.1: Security Audit Data Generation
	VPNGW12:FAU_GEN.1/VPN: Audit Data Generation (VPN Gateway)
	NDcPP22e:FAU_GEN.2: User identity association
	NDcPP22e:FAU_GEN_EXT.1: Security Audit Generation
	NDcPP22e:FAU_STG_EXT.1: Protected Audit Event Storage
	NDcPP22e:FAU_STG_EXT.4: Protected Local Audit Event Storage for Distributed TOEs
	NDcPP22e:FAU_STG_EXT.5: Protected Remote Audit Event Storage for Distributed TOEs
FCO: Communication	NDcPP22e:FCO_CPC_EXT.1: Component Registration Channel Definition
FCS: Cryptographic support	NDcPP22e:FCS_CKM.1: Cryptographic Key Generation
	VPNGW12:FCS_CKM.1/IKE: Cryptographic Key Generation (for IKE Peer Authentication)
	NDcPP22e:FCS_CKM.2: Cryptographic Key Establishment
	NDcPP22e:FCS_CKM.4: Cryptographic Key Destruction
	NDcPP22e:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption)
	VPNGW12:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption)
	NDcPP22e:FCS_COP.1/Hash: Cryptographic Operation (Hash Algorithm)
	NDcPP22e:FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm)
	NDcPP22e:FCS_COP.1/SigGen: Cryptographic Operation (Signature Generation and Verification)
	NDcPP22e:FCS_HTTPS_EXT.1: HTTPS Protocol
	NDcPP22e:FCS_IPSEC_EXT.1: IPsec Protocol - per TD0633
	VPNGW12:FCS_IPSEC_EXT.1: IPsec Protocol - per TD0657
	NDcPP22e:FCS_NTP_EXT.1: NTP Protocol

	NDcPP22e:FCS_RBG_EXT.1: Random Bit Generation
	NDcPP22e:FCS_TLSC_EXT.1: TLS Client Protocol Without Mutual Authentication - per TD0634
	NDcPP22e:FCS_TLSC_EXT.2: TLS Client Support for Mutual Authentication
	NDcPP22e:FCS_TLSS_EXT.1: TLS Server Protocol Without Mutual Authentication - per TD0635
	NDcPP22e:FCS_TLSS_EXT.2: TLS Server Support for Mutual Authentication
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	NDcPP22e:FIA_AFL.1: Authentication Failure Management
	NDcPP22e:FIA_PMG_EXT.1: Password Management
	NDcPP22e:FIA_UAU.7: Protected Authentication Feedback
	NDcPP22e:FIA_UAU_EXT.2: Password-based Authentication Mechanism
	NDcPP22e:FIA_UIA_EXT.1: User Identification and Authentication
	NDcPP22e:FIA_X509_EXT.1/ITT: X.509 Certificate Validation
	NDcPP22e:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
	VPNGW12:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
	NDcPP22e:FIA_X509_EXT.2: X.509 Certificate Authentication
	VPNGW12:FIA_X509_EXT.2: X.509 Certificate Authentication
	NDcPP22e:FIA_X509_EXT.3: X.509 Certificate Requests
	VPNGW12:FIA_X509_EXT.3: X.509 Certificate Requests
FMT: Security management	NDcPP22e:FMT_MOF.1/Functions: Management of Security Functions Behaviour
	NDcPP22e:FMT_MOF.1/ManualUpdate: Management of security functions behaviour
	NDcPP22e:FMT_MTD.1/CoreData: Management of TSF Data
	NDcPP22e:FMT_MTD.1/CryptoKeys: Management of TSF Data
	VPNGW12:FMT_MTD.1/CryptoKeys: Management of TSF Data
	NDcPP22e:FMT_SMF.1: Specification of Management Functions - per TD0631
	STFFW14e:FMT_SMF.1/FFW: Specification of Management Functions
	VPNGW12:FMT_SMF.1/VPN: Specification of Management Functions
	NDcPP22e:FMT_SMR.2: Restrictions on Security Roles
FPF: Packet Filtering	VPNGW12:FPF_RUL_EXT.1: Packet Filtering Rules
FPT: Protection of the TSF	NDcPP22e:FPT_APW_EXT.1: Protection of Administrator Passwords
	VPNGW12:FPT_FLS.1/SelfTest: Failure with Preservation of Secure State (Self-Test Failures)
	NDcPP22e:FPT_ITT.1: Basic internal TSF data transfer protection - per TD0639
	NDcPP22e:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
	NDcPP22e:FPT_STM_EXT.1: Reliable Time Stamps - per TD0632
	NDcPP22e:FPT_TST_EXT.1: TSF testing
	VPNGW12:FPT_TST_EXT.1: TSF Testing
	VPNGW12:FPT_TST_EXT.3: Self-Test with Defined Methods

	NDcPP22e:FPT_TUD_EXT.1: Trusted update
	VPNGW12:FPT_TUD_EXT.1: Trusted Update
FTA: TOE access	NDcPP22e:FTA_SSL.3: TSF-initiated Termination
	VPNGW12:FTA_SSL.3/VPN: TSF-Initiated Termination (VPN Headend) - per TD0656
	NDcPP22e:FTA_SSL.4: User-initiated Termination
	NDcPP22e:FTA_SSL_EXT.1: TSF-initiated Session Locking
	NDcPP22e:FTA_TAB.1: Default TOE Access Banners
	VPNGW12:FTA_TSE.1: TOE Session Establishment - per TD0656
	VPNGW12:FTA_VCM_EXT.1: VPN Client Management - per TD0656
FTP: Trusted path/channels	NDcPP22e:FTP_ITC.1: Inter-TSF trusted channel - per TD0639
	VPNGW12:FTP_ITC.1/VPN: Inter-TSF Trusted Channel (VPN Communications)
	NDcPP22e:FTP_TRP.1/Admin: Trusted Path - per TD0639
	NDcPP22e:FTP_TRP.1/Join: Trusted Path - per TD0639

Table 1 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit Data Generation (NDcPP22e/STFFW14e/:FAU_GEN.1)

NDcPP22e:FAU_GEN.1.1

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

a) Start-up and shut-down of the audit functions;

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

c) All administrative actions comprising:

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

d) Specifically defined auditable events listed in Table 2.

Requirement	Auditable Events	Additional Content
NDcPP22e:FAU_GEN.1	None	None
STFFW14e:FAU_GEN.1	None	None
VPNGW12:FAU_GEN.1/VPN	No events specified.	N/A
NDcPP22e:FAU_GEN.2	None	None
NDcPP22e:FAU_GEN_EXT.1	None	None
NDcPP22e:FAU_STG_EXT.1	None	None
NDcPP22e:FAU_STG_EXT.4	None	None
NDcPP22e:FAU_STG_EXT.5	None	None
NDcPP22e: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.

NDcPP22e:FCS_CKM.1	None	None
VPNGW12:FCS_CKM.1/IKE	No events specified.	N/A
NDcPP22e:FCS_CKM.2	None	None
NDcPP22e:FCS_CKM.4	None	None
NDcPP22e:FCS_COP.1/DataEncryption	None	None
VPNGW12:FCS_COP.1/DataEncryption	None	None
NDcPP22e:FCS_COP.1/Hash	None	None
NDcPP22e:FCS_COP.1/KeyedHash	None	None
NDcPP22e:FCS_COP.1/SigGen	None	None
NDcPP22e:FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure.
NDcPP22e:FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure.
NDcPP22e:FCS_NTP_EXT.1	Configuration of a new time server Removal of configured time server	Identity if new/removed time server
NDcPP22e:FCS_RBG_EXT.1	None	None
NDcPP22e:FCS_TLSC_EXT.1	Failure to establish a TLS Session.	Reason for failure.
NDcPP22e:FCS_TLSC_EXT.2	None	None
NDcPP22e:FCS_TLSS_EXT.1	Failure to establish a TLS Session.	Reason for failure.
NDcPP22e: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 Source and destination ports Transport Layer Protocol TOE Interface
STFFW14e:FFW_RUL_EXT.2	Dynamical definition of rule Establishment of a session	None
NDcPP22e:FIA_AFL.1	Unsuccessful login attempt limit is met or exceeded.	Origin of the attempt (e.g., IP address).
NDcPP22e:FIA_PMG_EXT.1	None	None
NDcPP22e:FIA_UAU.7	None	None
NDcPP22e:FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP22e:FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP22e: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
NDcPP22e: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
VPNGW12:FIA_X509_EXT.1/Rev	None	None
NDcPP22e:FIA_X509_EXT.2	None	None
VPNGW12:FIA_X509_EXT.2	None	None
NDcPP22e:FIA_X509_EXT.3	None	None
VPNGW12:FIA_X509_EXT.3	None	None
NDcPP22e:FMT_MOF.1/Functions	None	None

NDcPP22e:FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	None
NDcPP22e:FMT_MTD.1/CoreData	None	None
NDcPP22e:FMT_MTD.1/CryptoKeys	None	None
VPNGW12:FMT_MTD.1/CryptoKeys	None	None
NDcPP22e: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
VPNGW12:FMT_SMF.1/VPN	All administrative actions	No additional information.
NDcPP22e:FMT_SMR.2	None	None
VPNGW12:FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol
NDcPP22e:FPT_APW_EXT.1	None	None
VPNGW12:FPT_FLS.1/SelfTest	No events specified.	N/A
NDcPP22e: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.
NDcPP22e:FPT_SKP_EXT.1	None	None
NDcPP22e: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).
NDcPP22e:FPT_TST_EXT.1	None	None
VPNGW12:FPT_TST_EXT.1	None	None
VPNGW12:FPT_TST_EXT.3	No events specified.	N/A
NDcPP22e:FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	None
VPNGW12:FPT_TUD_EXT.1	None	None
NDcPP22e:FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None
VPNGW12:FTA_SSL.3/VPN	No events specified.	N/A
NDcPP22e:FTA_SSL.4	The termination of an interactive session.	None
NDcPP22e:FTA_SSL_EXT.1	(if 'lock the session' is selected) Any attempts at unlocking of an interactive session. (if 'terminate the session' is selected) The termination of a local session by the session locking mechanism.	None
NDcPP22e:FTA_TAB.1	None	None
VPNGW12:FTA_TSE.1	No events specified.	N/A
VPNGW12:FTA_VCM_EXT.1	No events specified.	N/A
NDcPP22e:FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted	Identification of the initiator and target of failed trusted channels

	channel. Failure of the trusted channel functions.	establishment attempt.
VPNGW12:FTP_ITC.1/VPN	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	No additional information. No additional information. Identification of the initiator and target of failed trusted channels establishment attempt
NDcPP22e:FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None
NDcPP22e:FTP_TRP.1/Join	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None

Table 2 Audit events

NDcPP22e/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) (VPNGW12:FAU_GEN.1/VPN)

VPNGW12:FAU_GEN.1/VPN.1

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

a. Start-up and shutdown of the audit functions
b. 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 the Auditable Events for Mandatory Requirements table.

VPNGW12:FAU_GEN.1/VPN.2

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

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

b. For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, additional information defined in the Auditable Events for Mandatory Requirements table for each auditable event, where applicable.

5.1.1.3 User identity association (NDcPP22e:FAU_GEN.2)

NDcPP22e: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 Generation (NDcPP22e:FAU_GEN_EXT.1)

NDcPP22e: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 (NDcPP22e:FAU_STG_EXT.1)

NDcPP22e: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.

NDcPP22e: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 NGFW Engine transfers audit records to the Virtual SMC Appliance].***]

NDcPP22e:FAU_STG_EXT.1.3

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 NGFW Engine, the NGFW Engine will stop traffic and transition into an offline state until audit space is again available]***] when the local storage space for audit data is full.

5.1.1.6 Protected Local Audit Event Storage for Distributed TOEs (NDcPP22e:FAU_STG_EXT.4)

NDcPP22e: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 NGFW Engine, the NGFW 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 (NDcPP22e:FAU_STG_EXT.5)

NDcPP22e: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 (NDcPP22e:FCO_CPC_EXT.1)

NDcPP22e: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.

NDcPP22e: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].

NDcPP22e: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 (NDcPP22e:FCS_CKM.1)

NDcPP22e:FCS_CKM.1.1

The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm: [
- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3,
- ECC schemes using 'NIST curves' [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4
- 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].].

5.1.3.2 Cryptographic Key Generation (for IKE Peer Authentication) (VPNGW12:FCS_CKM.1/IKE)

VPNGW12: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 186-4, 'Digital Signature Standard (DSS)', Appendix B.3 for RSA schemes,***
- FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4 for ECDSA schemes and implementing 'NIST curves' P-384 and [P-256, P521]]
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.

5.1.3.3 Cryptographic Key Establishment (NDcPP22e:FCS_CKM.2)

NDcPP22e:FCS_CKM.2.1

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [
- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1",
- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, 'Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography' (TD0581 applied),
- 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] (TD0580 applied)].

5.1.3.4 Cryptographic Key Destruction (NDcPP22e:FCS_CKM.4)

NDcPP22e: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 [a pseudo-random pattern using the TSF's RBG, zeroes]]
that meets the following: No Standard.

5.1.3.5 Cryptographic Operation (AES Data Encryption/Decryption) (NDcPP22e:FCS_COP.1/DataEncryption)

NDcPP22e: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, 192 bits, 256 bits***] that meet the following: AES as specified in ISO 18033-3, [***CBC as specified in ISO 10116, GCM as specified in ISO 19772***].

5.1.3.6 Cryptographic Operation (AES Data Encryption/Decryption) (VPNGW12:FCS_COP.1/DataEncryption)

VPNGW12: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) (NDcPP22e:FCS_COP.1/Hash)

NDcPP22e: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) (NDcPP22e:FCS_COP.1/KeyedHash)

NDcPP22e: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) (NDcPP22e:FCS_COP.1/SigGen)

NDcPP22e:FCS_COP.1.1/SigGen

The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [
- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [2048 bits, 3072 bits and 4096 bits],
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256, 384, 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 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,***
- For ECDSA schemes: FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Section 6 and Appendix D, Implementing 'NIST curves' [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4].

5.1.3.10 HTTPS Protocol (NDcPP22e:FCS_HTTPS_EXT.1)

NDcPP22e:FCS_HTTPS_EXT.1.1

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

NDcPP22e:FCS_HTTPS_EXT.1.2

The TSF shall implement HTTPS using TLS.

NDcPP22e:FCS_HTTPS_EXT.1.3

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

5.1.3.11 IPsec Protocol - per TD0633 (NDcPP22e:FCS_IPSEC_EXT.1)

NDcPP22e:FCS_IPSEC_EXT.1.1

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

NDcPP22e: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.

NDcPP22e:FCS_IPSEC_EXT.1.3

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

NDcPP22e: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***].

NDcPP22e:FCS_IPSEC_EXT.1.5

The TSF shall implement the protocol: [***- IKEv2 as defined in RFC 5996 and [with mandatory support for NAT traversal as specified in RFC 5996, section 2.23], and [RFC 4868 for hash functions]***].

NDcPP22e: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)***].

NDcPP22e:FCS_IPSEC_EXT.1.7

The TSF shall ensure that [***- IKEv2 SA lifetimes can be configured by a Security Administrator based on [length of time, where the time values can be configured within [24] hours]***].

NDcPP22e: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 within [8] hours]].

NDcPP22e: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 [**twice the security strength of the negotiated Diffie-Hellman group**] bits.

NDcPP22e: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***].

NDcPP22e: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]***].

NDcPP22e: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.

NDcPP22e: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***].

NDcPP22e: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 TD0657 (VPNGW12:FCS_IPSEC_EXT.1)

VPNGW12:FCS_IPSEC_EXT.1.1

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

VPNGW12: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.

VPNGW12:FCS_IPSEC_EXT.1.3

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

VPNGW12: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***].

VPNGW12:FCS_IPSEC_EXT.1.5

The TSF shall implement the protocol: [***IKEv2 as defined in RFC 5996 and [with mandatory support for NAT traversal as specified in RFC 5996, section 2.23] and [RFC 4868 for hash functions]***].

VPNGW12: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)***].

VPNGW12:FCS_IPSEC_EXT.1.7

The TSF shall ensure that [***IKEv2 SA lifetimes can be configured by a Security Administrator based on [length of time, where the time values can be configured within [24] hours]***].

VPNGW12: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 within [8] hours]***].

VPNGW12: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 [**128-bits that is at least twice the security strength of the negotiated DH group**] bits.

VPNGW12:FCS_IPSEC_EXT.1.10

The TSF shall generate nonces used in [***IKEv2***] exchanges of length [***128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash***].

VPNGW12:FCS_IPSEC_EXT.1.11

The TSF shall ensure that IKE protocols implement DH Group(s)
- 19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and
- [***[selection: 14 (2048-bit MODP), 15 (3072-bit MODP), 16 (4096-bit MODP), 17 (6144-bit MODP), 18 (8192-bit MODP)] according to RFC 3526, [21 (521-bit Random ECP)] according to RFC 5114***].

VPNGW12: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.

VPNGW12: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***].

VPNGW12: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 (NDcPP22e:FCS_NTP_EXT.1)

NDcPP22e:FCS_NTP_EXT.1.1

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

NDcPP22e:FCS_NTP_EXT.1.2

The TSF shall update its system time using [***Authentication using [SHA1] as the message digest algorithm(s);***].

NDcPP22e:FCS_NTP_EXT.1.3

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

NDcPP22e: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 (NDcPP22e:FCS_RBG_EXT.1)

NDcPP22e:FCS_RBG_EXT.1.1

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

NDcPP22e:FCS_RBG_EXT.1.2

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

5.1.3.15 TLS Client Protocol Without Mutual Authentication - per TD0634 (NDcPP22e:FCS_TLSC_EXT.1)

NDcPP22e:FCS_TLSC_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: [

<u>**Syslog over TLS (SMC as Client)**</u>
TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,
TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,
TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288,
TLS_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_CBC_SHA384 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289

] and no other ciphersuites.

NDcPP22e: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 SAN***].

NDcPP22e:FCS_TLSC_EXT.1.3
When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [***Not implement any administrator override mechanism***].

NDcPP22e:FCS_TLSC_EXT.1.4
The TSF shall [***present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups: [secp256r1, secp384r1, secp521r1] and no other curves/groups***] in the Client Hello.

5.1.3.16 TLS Client Support for Mutual Authentication (NDcPP22e:FCS_TLSC_EXT.2)

NDcPP22e: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 Without Mutual Authentication - per TD0635 (NDcPP22e:FCS_TLSS_EXT.1)

NDcPP22e: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_CBC_SHA384 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
]
and no other ciphersuites.

NDcPP22e:FCS_TLSS_EXT.1.2
The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [***TLS 1.1***].

NDcPP22e:FCS_TLSS_EXT.1.3
The TSF shall perform key establishment for TLS using [***ECDHE curves [secp256r1, secp384r1, secp521r1] and no other curves***]].

NDcPP22e:FCS_TLSS_EXT.1.4
The TSF shall support [***no session resumption or session tickets***].

5.1.3.18 TLS Server Support for Mutual Authentication (NDcPP22e:FCS_TLSS_EXT.2)

NDcPP22e:FCS_TLSS_EXT.2.1

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

NDcPP22e: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***].

NDcPP22e: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.

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 (NDcPP22e:FIA_AFL.1)

NDcPP22e: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.

NDcPP22e: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 (NDcPP22e:FIA_PMG_EXT.1)

NDcPP22e: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 Protected Authentication Feedback (NDcPP22e:FIA_UAU.7)

NDcPP22e: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.

5.1.6.4 Password-based Authentication Mechanism (NDcPP22e:FIA_UAU_EXT.2)

NDcPP22e:FIA_UAU_EXT.2.1

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

5.1.6.5 User Identification and Authentication (NDcPP22e:FIA_UIA_EXT.1)

NDcPP22e: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]***].

NDcPP22e:FIA_UIA_EXT.1.2

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

5.1.6.6 X.509 Certificate Validation (NDcPP22e:FIA_X509_EXT.1/ITT)

NDcPP22e:FIA_X509_EXT.1/ITT.1

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:
  - o 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
  - o 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
  - o 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.

NDcPP22e:FIA_X509_EXT.1/ITT.2

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

5.1.6.7 X.509 Certificate Validation (NDcPP22e:FIA_X509_EXT.1/Rev)

NDcPP22e:FIA_X509_EXT.1/Rev.1

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:
  - o 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.
  - o 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.
  - o 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.
  - o 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.

NDcPP22e:FIA_X509_EXT.1/Rev.2

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

5.1.6.8 X.509 Certificate Validation (VPNGW12:FIA_X509_EXT.1/Rev)

VPNGW12:FIA_X509_EXT.1/Rev.1

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:
  - o 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.
  - o 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.
  - o 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.
  - o 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.

5.1.6.9 X.509 Certificate Authentication (NDcPP22e:FIA_X509_EXT.2)

NDcPP22e: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***].

NDcPP22e: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***].

5.1.6.10 X.509 Certificate Authentication (VPNGW12:FIA_X509_EXT.2)

VPNGW12: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***].

VPNGW12:FIA_X509_EXT.2.2

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

5.1.6.11 X.509 Certificate Requests (NDcPP22e:FIA_X509_EXT.3)

NDcPP22e: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***].

NDcPP22e: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 (NDcPP22e:FMT_MOF.1/Functions)

NDcPP22e:FMT_MOF.1/Functions.1

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 (NDcPP22e:FMT_MOF.1/ManualUpdate)

NDcPP22e:FMT_MOF.1/ManualUpdate.1

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 (NDcPP22e:FMT_MTD.1/CoreData)

NDcPP22e:FMT_MTD.1/CoreData.1

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

5.1.7.4 Management of TSF Data (NDcPP22e:FMT_MTD.1/CryptoKeys)

NDcPP22e:FMT_MTD.1/CryptoKeys.1

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

5.1.7.5 Management of TSF Data (VPNGW12:FMT_MTD.1/CryptoKeys)

VPNGW12:FMT_MTD.1/CryptoKeys.1

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 TD0631 (NDcPP22e:FMT_SMF.1)

NDcPP22e:FMT_SMF.1.1

The TSF shall be capable of performing the following management functions:
- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [***digital signature***] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [***Ability to configure audit behavior (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full),***

Ability to configure the list of TOE-provided services available before an entity is identified and authenticated, as specified in FIA_UIA_EXT.1,
Ability to manage the cryptographic keys, Ability to configure the cryptographic functionality,
Ability to configure the lifetime for IPsec SAs,
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 import X509v3 certificates to the TOE's trust store].

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

STFFW14e:FMT_SMF.1/FFW.1

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

5.1.7.8 Specification of Management Functions (VPNGW12:FMT_SMF.1/VPN)

VPNGW12:FMT_SMF.1/VPN.1

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 (NDcPP22e:FMT_SMR.2)

NDcPP22e:FMT_SMR.2.1
The TSF shall maintain the roles: - Security Administrator.

NDcPP22e:FMT_SMR.2.2
The TSF shall be able to associate users with roles.

NDcPP22e:FMT_SMR.2.3
The TSF shall ensure that the conditions
- The Security Administrator role shall be able to administer the TOE locally;
- The Security Administrator role shall be able to administer the TOE remotely
are satisfied.

5.1.8 Packet Filtering (FPF)

5.1.8.1 Packet Filtering Rules (VPNGW12:FPF_RUL_EXT.1)

VPNGW12:FPF_RUL_EXT.1.1
The TSF shall perform Packet Filtering on network packets processed by the TOE.

VPNGW12: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)
  o source address
  o destination address
  o protocol
- IPv6 (RFC 8260)
  o source address
  o destination address
  o next Header (protocol)
- TCP (RFC 793)
  o source port
  o destination port
- UDP (RFC 768)
  o source port
  o destination port.

(TD0683 applied)

VPNGW12: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.

VPNGW12:FPF_RUL_EXT.1.4
The TSF shall allow the Packet Filtering rules to be assigned to each distinct network interface.

VPNGW12:FPF_RUL_EXT.1.5
The TSF shall process the applicable Packet Filtering rules (as determined in accordance with VPNGW12:FPF_RUL_EXT.1.4) in the following order: Administrator-defined.

VPNGW12: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 (NDcPP22e:FPT_APW_EXT.1)

NDcPP22e:FPT_APW_EXT.1.1
The TSF shall store administrative passwords in non-plaintext form.

NDcPP22e: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) (VPNGW12:FPT_FLS.1/SelfTest)

VPNGW12:FPT_FLS.1/SelfTest.1
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 - per TD0639 (NDcPP22e:FPT_ITT.1)

NDcPP22e: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 pre-shared, symmetric and private keys) (NDcPP22e:FPT_SKP_EXT.1)

NDcPP22e: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 - per TD0632 (NDcPP22e:FPT_STM_EXT.1)

NDcPP22e:FPT_STM_EXT.1.1
The TSF shall be able to provide reliable time stamps for its own use.

NDcPP22e:FPT_STM_EXT.1.2
The TSF shall [***allow the Security Administrator to set the time, synchronise time with an NTP server***].

5.1.9.6 TSF testing (NDcPP22e:FPT_TST_EXT.1)

NDcPP22e:FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests [***during initial start-up (on power on)***] to demonstrate the correct operation of the TSF: [**KAT tests, and checksums of TOE binaries**].

5.1.9.7 TSF Testing (VPNGW12:FPT_TST_EXT.1)

VPNGW12:FPT_TST_EXT.1.1
The TSF shall run a suite of the following self-tests [***during initial startup (on power on)***] to demonstrate the correct operation of the TSF: noise source health tests, [**KAT tests, and checksums of TOE binaries**].

5.1.9.8 Self-Test with Defined Methods (VPNGW12:FPT_TST_EXT.3)

VPNGW12: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.

VPNGW12: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 (NDcPP22e:FPT_TUD_EXT.1)

NDcPP22e:FPT_TUD_EXT.1.1

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

NDcPP22e:FPT_TUD_EXT.1.2

The TSF shall provide Security Administrators the ability to manually initiate updates to TOE firmware/software and [***support automatic checking for updates***].

NDcPP22e: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 (VPNGW12:FPT_TUD_EXT.1)

VPNGW12:FPT_TUD_EXT.1.1

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

VPNGW12: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***].

VPNGW12: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.

5.1.10 TOE access (FTA)

5.1.10.1 TSF-initiated Termination (NDcPP22e:FTA_SSL.3)

NDcPP22e: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) - per TD0656 (VPNGW12:FTA_SSL.3/VPN)

VPNGW12:FTA_SSL.3/VPN.1

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 (NDcPP22e:FTA_SSL.4)

NDcPP22e: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 (NDcPP22e:FTA_SSL_EXT.1)

NDcPP22e: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 (NDcPP22e:FTA_TAB.1)

NDcPP22e: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 - per TD0656 (VPNGW12:FTA_TSE.1)

VPNGW12: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 - per TD0656 (VPNGW12:FTA_VCM_EXT.1)

VPNGW12: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 - per TD0639 (NDcPP22e:FTP_ITC.1)

NDcPP22e: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.

NDcPP22e:FTP_ITC.1.2

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

NDcPP22e:FTP_ITC.1.3

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

5.1.11.2 Inter-TSF Trusted Channel (VPN Communications) (VPNGW12:FTP_ITC.1/VPN)

VPNGW12:FTP_ITC.1/VPN.1

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.

VPNGW12:FTP_ITC.1.2

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

VPNGW12:FTP_ITC.1.3

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

5.1.11.3 Trusted Path - per TD0639 (NDcPP22e:FTP_TRP.1/Admin)

NDcPP22e:FTP_TRP.1/Admin.1

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.

NDcPP22e:FTP_TRP.1/Admin.2

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

NDcPP22e:FTP_TRP.1/Admin.3

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 - per TD0639 (NDcPP22e:FTP_TRP.1/Join)

NDcPP22e:FTP_TRP.1/Join.1

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***].

NDcPP22e:FTP_TRP.1/Join.2

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

NDcPP22e:FTP_TRP.1/Join.3

The TSF shall require the use of the trusted path for joining components to the TSF under environmental constraints identified in [**the Forcepoint Next Generation Firewall 6.10.9 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
	AVA_VLA.1: Additional Flaw Hypotheses

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 categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4c

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

ADV_FSP.1.1e

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

ADV_FSP.1.2e

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

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational User Guidance (AGD_OPE.1)

AGD_OPE.1.1d

The developer shall provide operational user guidance.

AGD_OPE.1.1c

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

AGD_OPE.1.2c

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

AGD_OPE.1.3c

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

AGD_OPE.1.4c

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

AGD_OPE.1.5c

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

AGD_OPE.1.6c

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill 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 NGFW Engine generates audit events pertaining to packet filtering. All other security relevant audit events are generated by all TOE components.

The NGFW 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.

When the NGFW Engine cannot transfer newly generated audit records to the Log Server (irrespective of the cause), the Engine will spool the records on disk. Once the NGFW Engine can again transfer audit records to the Log Server, it transfers the oldest records first and removes those from its spool, continuing until it has transferred all spooled records. If the NGFW Engine cannot transfer its audit records to the Log Server for an extended period of time, it may start to exhaust its available spool disk storage space (the size of this spool depends on the size of the disk in the NGFW Engine model, but the N120/N120W/N120WL/N120L/N60/N60L/2201/2205/2210/3401/3405/3410/Virtual models provide 3/3/8/8/3/3/48/48/48/201/201/201/24 GB of spool space, respectively). In such an event the NGFW Engine first sends alerts to notify the administrator that it has nearly exhausted its log spool, and once it exhausts its spool space, it follows the administrator defined behavior.

The Log Server, which aggregates audit records from NGFW Engines, writes incoming audit entries to the 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 NGFW 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.

As mentioned above, the administrator defines the log spool policy for the NGFW Engines. This specifies the behavior of the NGFW Engines whenever its local log spool fills. The TOE supports two settings, but requires that only the following be used when in an evaluated configuration:

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

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

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

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

The Security audit function satisfies the following security functional requirements:

- NDcPP22e/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**.
- NDcPP22e: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.

- NDcPP22e:FAU_GEN_EXT.1: The NGFW Engines and the Virtual SMC Appliance each generate audit records specific to the component's functionality. NGFW Engines primarily generate firewall and TLS secure channel audits while the Virtual SMC Appliance generates TLS secure channel audits and audits of administrative actions.
- NDcPP22e:FAU_STG_EXT.1/4/5: All audit records generated by the NGFW 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). The administrator can define NGFW Engine behavior should it lose connectivity with the Virtual SMC Appliance and exhaust the local storage audit space. The Virtual SMC Appliance, should it exhaust its audit log storage space, halts generation of new audit data.
- VPNGW12: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 "VPNGW12"), 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:

- NDcPP22e: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 NGFW 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 NGFW Engine and the SMC via the SMC Client GUI. This is done from the perspective of the NGFW Engine by performing a factory reset of the Engine and from the perspective of the SMC by deleting the NGFW 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. NGFW Engine
   a. NGFW Engine uses cryptography from the Forcepoint NGFW FIPS Cryptographic Module 1.2.1 (based upon SafeZone FIPS Cryptographic Module) for IKE connections, TLS connections, code integrity, and trusted updates.

b. NGFW Engine uses cryptography from the Forcepoint NGFW Cryptographic Kernel Module 3.0 for IPsec connections.

2. Virtual SMC Appliance
    a. Virtual SMC Appliance uses cryptography from its SMC FIPS Library 1.1.1 based on OpenSSL for password hashing, integrity testing and verification of Trusted Updates
    b. Virtual SMC Appliance's Management and Log Server use cryptography from the SMC FIPS Java API 1.0.2.3 for the TOE TLS, CSR generation, and password hashing
    c. Virtual SMC Appliance's NTP daemon uses cryptography from its SMC FIPS Cryptographic Module for NTP 3.79

Version 6.10.9 of the NGFW Engine internally includes version 10 of the NGFW OS, which itself includes a Linux 4.19 kernel.

SFR	Algorithm	NIST/ISO Standard	Cert#
FCS_CKM.1 (Key Gen)	RSA IFC Key Generation 2048/3072/4096 bit	FIPS 186-4, RSA	A2155
	ECDSA ECC Key Generation P-256, P-384, P-521	FIPS 186-4, ECDSA	A2155
FCS_CKM.2 (Key Establishment)	RSA-based Key Exchange	Vendor affirm RSAES-PKCS1-v1_5	N/A
	ECC-based Key Exchange P-256, P-384, P-521	SP 800-56A, KAS ECC	A2155
FCS_COP.1/DataEncryption	AES CBC - 128/192/256 AES GCM - 128/256	ISO 10116 ISO 19772	A2155
FCS_COP.1/SigGen	ECDSA Sign/Verify P-256, P-384, P-521	FIPS 186-4, ECDSA	A2155
	RSA Sign/Verify 2048, 3072, 4096	FIPS 186-4, RSA	A2155
FCS_COP.1/Hash	SHA Hashing SHA-1, SHA-256, SHA-384, SHA-512	ISO/IEC 10118-3:2004	A2155
FCS_COP.1/KeyedHash	Keyed Hash HMAC-SHA-1 HMAC-SHA-256, HMAC-SHA-384	ISO/IEC 9797-2:2011	A2155
FCS_RBG_EXT.1 (CTR) (Random)	CTR_DRBG (AES) 256 bits	ISO/IEC 18031:2011	A2155
FCS_IPSEC	IKEv2 KDF	SP 800-135	A2155
FCS_TLS	TLS KDF	SP 800-135	A2155

Table 4 Forcepoint NGFW FIPS Cryptographic Module 1.2.1 (based upon SafeZone FIPS Cryptographic Module) CAVP Certificates

SFR	Algorithm	NIST/ISO Standard	Cert#
FCS_COP.1/DataEncryption	AES 128/192/256 CBC, GCM	ISO 10116 ISO 19772	A2166
FCS_COP.1/Hash	SHA Hashing SHA-1, SHA-256, SHA-384, SHA-512	ISO/IEC 10118-3:2004	A2166
FCS_COP.1/KeyedHash	Keyed Hash HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	ISO/IEC 9797-2:2011	A2166

Table 5 Forcepoint NGFW Cryptographic Kernel Module 3.0 CAVP Certificates

The TOE includes the SMC FIPS Java API 1.0.2.3 library which was deemed equivalent to the 1.0.2.1 version of the library as they differ only because of non-security relevant changes. These non-security relevant changes include a fix for JVM class loader issues manifesting in the October 2020 updates to Java 11, fixes to remove two performance bottlenecks, and a change to allow configuring the library's system properties though java.security. These non-security relevant changes were reviewed by a laboratory and the CMVP and the different versions of the library were added to the library's FIPS 140-2 validation certificate.

SFR	Algorithm	NIST Standard	Cert#
FCS_CKM.1 (Key Gen)	RSA IFC Key Generation 2048/3072 bit	FIPS 186-4, RSA	A1973
	ECDSA ECC Key Generation P-256, P-384, P-521	FIPS 186-4, ECDSA	A1973
FCS_CKM.2 (Key Establishment)	RSA-based Key Exchange	Vendor affirm RSAES-PKCS1-v1_5	N/A
	ECC-based Key Exchange P-256, P-384, P-521	SP 800-56A, KAS ECC	A1973
FCS_COP.1/DataEncryption	AES 128/256 CBC, GCM	ISO 10116 ISO 19772	A1973
FCS_COP.1/SigGen	RSA Sign/Verify 2048/3072 bit	FIPS 186-4, RSA	A1973
	ECDSA Sign/Verify P-256, P-384, P-521	FIPS 186-4, ECDSA	A1973
FCS_COP.1/Hash	SHA Hashing SHA-1, SHA-256, SHA-384, SHA-512	ISO/IEC 10118=3:2004	A1973
FCS_COP.1/KeyedHash	Keyed Hash HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384	ISO/IEC 9797-2:2011	A1973
FCS_RBG_EXT.1 (CTR, Hash)(Random)	DRBG Bit Generation CTR_DRBG (AES) -256 bits Hash_DRBG (SHA-512)	ISO/IEC 18031:2011	A1973

Table 6 Virtual SMC Appliance SMC FIPS Java API 1.0.2.3 CAVP Certificates

Version 6.10.9 of the Virtual SMC Appliance internally includes an ESXi 7.0 hypervisor, running a RHEL 8 OS virtual machine. The RHEL 8 OS (running as a virtual machine) includes a Linux 4.18 kernel.

SFR	Algorithm	NIST Standard	Cert#
FCS_CKM.1 (Key Gen)	ECDSA ECC Key Generation P-256, P-384, P-521	FIPS 186-4, ECDSA	A1972
FCS_CKM.2 (Key Establishment)	ECC-based Key Exchange P-256, P-384, P-521	SP 800-56A, KAS ECC	A1972
FCS_COP.1/DataEncryption	AES 128/256 CBC, GCM	ISO 10116 ISO 19772	A1972
FCS_COP.1/SigGen	ECDSA Sign/Verify P-256, P-384, P-521	FIPS 186-4, ECDSA	A1972
FCS_COP.1/Hash	SHA Hashing SHA-1, SHA-256, SHA-384, SHA-512	ISO/IEC 10118-3:2004	A1972
FCS_COP.1/KeyedHash	Keyed Hash HMAC-SHA-256, HMAC-SHA-384	ISO/IEC 9797-2:2011	A1972
FCS_RBG_EXT.1(CTR) (Random)	DRBG Bit Generation CTR_DRBG (AES)- 256 bits	ISO/IEC 18031:2011	A1972

Table 7 Virtual SMC Appliance SMC FIPS Library 1.1.1 based on OpenSSL CAVP Certificates

SFR	Algorithm	NIST Standard	Cert#
FCS_COP.1/Hash	SHA Hashing, SHA-1	ISO/IEC 10118-3:2004	A3360

Table 8 Virtual SMC Appliance SMC FIPS Cryptographic Module for NTP 3.79 CAVP Certificates

6.3.1 NGFW Engine

The NGFW Engine directly calls the Forcepoint NGFW FIPS Cryptographic Module 1.2.1 (based upon SafeZone FIPS Cryptographic Module) for hashing, HMAC, encryption/decryption, and signature related operations supporting IKE connections (again see **Table 4**).

The Engine also utilizes this same cryptographic module as an OpenSSL engine plugin to provide cryptography for the Engine's OpenSSL 1.1.1 based Forcepoint NGFW FIPS Library 1.1.1. The Engine uses this library for

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

Finally, the NGFW Engine uses the Forcepoint NGFW Cryptographic Kernel Module 3.0 to provide hashing, HMAC, and encryption/decryption operations supporting IPsec/ESP connections (see **Table 5**).

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, and the cryptography verifying the validity of the update comes from the Virtual SMC Appliance SMC FIPS Library 1.1.1 based on OpenSSL.

The Virtual SMC Appliance SMC FIPS Library 1.1.1 based on OpenSSL 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 the Linux kernel Random Number Generator (LKRNG) to provide output to user space (through /dev/random). The Virtual SMC Appliance uses /dev/random to instantiate its SMC FIPS Java API 1.0.2.3's SHA-512 HASH_DRBG and generate keys. The SMC FIPS Java API 1.0.2.3 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 SMC FIPS Java API 1.0.2.3 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) as described in 6.3.

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 9.

The Management Server also accepts incoming administrative sessions (SMC Client connections) that are protected by TLS, and uses the cipher suites identified by Table 10 below.

6.3.2.2 Log Server

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

The Log Server utilizes the SMC FIPS Java API 1.0.2.3 Library for all encryption, decryption, hashing and signature operations associated with support for the TLS protocol.

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

TLS_RSA_WITH_AES_128_CBC_SHA
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_ SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_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_RSA_WITH_AES_128_GCM_SHA256
TLS_RSA_WITH_AES_256_GCM_SHA384
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 9 Cipher suites to communicate with an External Syslog Server

The SMC FIPS Java API 1.0.2.3 used by the Virtual SMC Appliance (and thus used by both the Management and Log Servers) uses the same /dev/random to instantiate its SMC FIPS Java API 1.0.2.3'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 10 Cipher suites to communicate with remote administrators

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

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

Table 11 Cipher suite for distributed TOE communication

Note that the ciphersuite identified in Table 10 above is the only one supported by both the SMC and the NGFW Engine for distributed TOE communication. The SMC supports the following additional ciphersuites for distributed TOE communication which were tested for completeness:

SMC (as both Client and Server):

- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

SMC (as Client):

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

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
ESP SA session keys	In Memory	SA deleted	Overwriting with zeros
Passwords	On Disk	Command	Overwriting once with pseudo-random data

Table 12 CSPs and Keys

The Cryptographic support function satisfies the following security functional requirements:

- NDcPP22e: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 or ECDH keys used during ECDHE_* and TLS_RSA_* TLS cipher suite negotiation.

TOE Component	Client or Server?	DH key gen?	ECDH gen?	ECDSA gen?	RSA gen?
Mgmt Server	Both	N/A	Yes	Yes	Yes
Log Server	Both	N/A	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] key to use for TLS Client or mutual authentication during TLS syslog export).

- VPNGW12: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 186-4) used during IKEv2 negotiation.

TOE Component	Performs IKE?	ECDSA gen?	RSA gen?	FFC 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 186-4 Appendix B.3.3 (Generation of Random Primes that are Probably Prime) and B.3.6 (Generation of Probably Primes with Conditions Based on Auxiliary Probably Primes) and ECDSA key generation per FIPS 186-4 Appendix B.4.2 (Key Pair Generation by Testing Candidates). 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

- NDcPP22e:FCS_CKM.2: the following table identifies the supported key establishment schemes mapped to the associated SFRs and their usage.

Scheme	SFR	Services
RSA	FCS_TLSC_EXT.1 FCS_TLSC_EXT.2	Syslog over TLS
ECC (P-256/384/521)	FCS_TLSC_EXT.1 FCS_TLSC_EXT.2	Syslog over TLS Distributed TOE Communication
	FCS_TLSS_EXT.1	HTTPS/TLS Remote Administration Distributed TOE Communication
	FCS_TLSS_EXT.2	Distributed TOE Communication
	FCS_IPSEC_EXT.1.11	IKEv2 key exchange
FCC (DH-14/15/16/17/18)	FCS_IPSEC_EXT.1.11	IKEv2 key exchange

- NDcPP22e: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 SMC FIPS Java API. After use on the NGFW Engine, keys are overwritten with zeros. This is performed by the NGFW 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 NGFW 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.

- NDcPP22e/VPNGW12: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**.

- NDcPP22e: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.

- NDcPP22e: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.

- NDcPP22e: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.

- NDcPP22e: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.

- NDcPP22e/VPNGW12: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) 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.

  The TOE supports DH groups 14 through 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.

- NDcPP22e: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.

- NDcPP22e:FCS_RBG_EXT.1: The TOE components perform random bit generation in support of the cryptographic functions. The SMC uses both an AES-256 CTR_DRBG and a SHA-512 Hash_DRBG while the Engines use an AES-256 CTR_DRBG. The TOE components use a software-based noise source and the SMC draws from /dev/random to instantiate both the CTR_DRBGs and Hash_DRBG while the Engines draw directly from the software-based noise source when instantiating its DRBG.

- NDcPP22e: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 or IPv4 address. 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. When configured with an IPv4 address, the TOE will check the IPv4 address against the certificate's SAN identifier field (the TOE does not support checking of an IPv4 address contained with the CN, only with the SAN). For communication with distributed TOE components, the TOE mandates a numerical identifier to be found in the SAN:DNS extension. This expected numerical identifier is negotiated at the time of registration of the NGFW 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.

- NDcPP22e:FCS_TLSS_EXT.1/2: The TOE provides a TLS interface for GUI administration. The TOE's TLS server acts similarly to its TLS client in that the administrator need not (and cannot) explicitly configure anything regarding the ECDHE curves, the TOE will negotiate P-256, P-384, or P-521 curves based upon what the peer/client specifies it supports in its hello. Similarly, the administrator need not and cannot specify the versions of TLS that the TOE's server will negotiate, the TOE only negotiates TLS v1.2 with clients. 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 TOE's different TLS server interfaces use the same cipher suites and curves. 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 TOE does not support session resumption or session tickets.

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 NGFW Engine enforces the firewall security policy on all traffic that passes through the engine, via its internal or external network Ethernet interfaces. The traffic is TCP, UDP, ICMPv4, ICMPv6, connections over IPv4 and IPv6. The NGFW Engine inspects and filters these protocols based upon their header fields as defined by their corresponding RFC (See Table 13).

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

Protocol	Related RFC[2]	Fields Inspected
ICMPv4	RFC 792	Type, Code

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

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 13 Protocols & Fields Filtered by the TOE

Any network traffic passed by the NGFW Engine must be explicitly allowed by a firewall rule or be part of an established session allowed by a rule, or it is dropped. This is true even in the case of attempts to flood a TOE interface (in which case some packets may be dropped, but no packets are ever passed that would violate policy).

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

The NGFW Engine has been designed to ensure that no residual information exists in network packets. When the NGFW Engine allocates a new buffer for either an incoming or outgoing network packet, the new packet data will be used to overwrite any previous data in the buffer. If an allocated buffer exceeds the size of the packet, any additional space will be overwritten (padded) with zeros before the packet is forwarded (to the external network or delivered to the appropriate, internal application). The NGFW Engine implements connection tracking to manage the information flow control decisions for connections (i.e., stateful sessions) rather than packets, providing increased performance and support for firewall features that require packet information above the IP level (e.g., ICMP, TCP, UDP). The connection tracking mechanism stores the state information of each connection to allow packets belonging to an established connection to pass. The connection tracking uses the fields shown in the following table when determining whether a packet matches an allowed established session for the corresponding protocol. Connection tracking follows the standard TCP handshaking process (SYN, responding SYN-ACK, followed by ACK) to denote establishment of a stateful session, and the TOE’s connection tracking will eliminate existing connections immediately, upon completion of the flow (in the case of TCP and FTP) or upon an inactivity timeout for the session.

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 14 Connection Tracking Fields

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

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

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

Each rule comprises matching criteria and target actions. If the matching criteria is verified (i.e., a comparison matches) the NGFW Engine applies the target actions. Possible target actions include Allow, Discard and Refuse[3]. Access rules with the logging option, can create a log or alert entry each time they match. The logging option is in addition to the target action of a rule. An administrator can specify that a rule applies to a specific interface by specifying a zone, adding a given firewall interface to that zone, and then specifying that zone as either a source or destination address for the rule.

The NGFW Engine compares the information attributes defined in Table 13 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 15 Additional Stateful Filtering Rules are applied.

Rules relating to FFW_RUL_EXT.1.6

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

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

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

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

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

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

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

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

Rules relating to FFW_RULE_EXT.1.7

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

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

c) The NGFW 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 15 Additional Stateful Filtering Rules

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

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

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

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

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

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

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

The 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 NGFW Engine filters network traffic using a rule base that comprises a set of security policy rules. These rules allow for complex security policies to be defined which control the flow of network traffic through the NGFW Engine. Controlled network traffic includes at least IPv4, IPv6, ICMPv4, ICMPv6, TCP and UDP protocols. Additional features of the firewall functionality are described above. The rule base is read from top down, and when the first matching rule is encountered the search stops and the TOE executes the matching rule. Any traffic that is not explicitly accepted by the security policy is rejected by the firewall.
- 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 NGFW Engine prior to an administrative user being authenticated. The TOE requires login prior to allowing any TOE configuration actions.

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

- NDcPP22e: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.

- NDcPP22e: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.
- Administrators can specify minimum lengths for passwords with 10 characters being the minimum in the evaluated configuration, and passwords can be no greater than 80 characters.
- NDcPP22e:FIA_UAU.7: All passwords entered by administrators are obscured when entered.
- NDcPP22e:FIA_UAU_EXT.2: The TOE authenticates administrative users by means of a local password mechanism.
- NDcPP22e: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.
- NDcPP22e/VPNGW12: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. The TOE does not support revocation for internal TOE communications between distributed TOE components. 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).
- NDcPP22e/VPNGW12:FIA_X509_EXT.2: 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.
- NDcPP22e: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 NGFW Engine certificates are issued during the registration process when the SMC sends the internal CA certificate to the NGFW 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 NGFW 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 NGFW 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:

- NDcPP22e:FMT_MOF.1/Functions: The TOE allows only authenticated administrators to configure TLS protected syslog export and to configure the handling of audit data if local audit storage space becomes full.
- NDcPP22e: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.
- NDcPP22e: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.
- NDcPP22e/VPNGW12:FMT_MTD.1/CryptoKeys: The TOE allows only authenticated administrators to configure (import, generate, delete, change) cryptographic keys.
- NDcPP22e/VPNGW12: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 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.
- NDcPP22e: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:

- VPNGW11: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 NGFW Engine and the servers running on the Virtual SMC Appliance are all protected using TLS network connections.

None of the TOE components utilize pre-shared keys or long-lived symmetric keys. The only keys retained by the components of the TOE are associated with certificates used for TLS. The servers running on the Virtual SMC Appliance store private keys in a password protected Java keystore.

Every appliance that is included as part of the TOE (i.e., NGFW Engines and Virtual SMC Appliance) includes its own real-time hardware-based clock. The time values from this clock are used in audit records. The NGFW Engine receives its time updates from the SMC Management Server or from configured NTP servers. The SMC Management Server is responsible for accepting and propagating clock updates initiated by an administrator. The NGFW Engine receives its time from the SMC unless the administrator has configured NTP.

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 NGFW FIPS Library 1.1.1 based upon OpenSSL 1.1.1 (which utilizes the Forcepoint NGFW FIPS Cryptographic Module 1.2.1), Virtual SMC Appliance SMC FIPS Library 1.1.1 based on OpenSSL, Virtual SMC Appliance SMC FIPS Java API 1.0.2.3, and SMC FIPS Cryptographic Module for NTP 3.79 cryptographic libraries. These KAT tests cover operation of AES, RSA, ECDSA, DRBG, SHA and HMAC-SHA. For each KAT test, the TOE uses known data as inputs into each cryptographic function, computes a cryptographic result (e.g., the AES ciphertext or SHA-512 hash), and compares the calculated result to the expected/known value. If the two do not match, the NGFW Engine will reboot as a result of the error, while the 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 NGFW Engine (using its Forcepoint NGFW FIPS Library 1.1.1 based upon OpenSSL 1.1.1 (which utilizes the Forcepoint NGFW FIPS Cryptographic Module 1.2.1) uses a preinstalled public key to verify the ECDSA P-521 with SHA-512 signature of the whole partition containing 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 NGFW Engines. To update the TOE software of the NGFW Engine, an administrator can obtain an update from Forcepoint and then upload the update to the SMC. After the SMC has the update, the SMC will verify the Forcepoint ECDSA P-521 w/ SHA-512 signature on the update package and, only if the signature verifies correctly, the SMC will import that package, making it available to update administrator-specified NGFW Engines with the new software. Once the administrator selects to upgrade a specific NGFW Engine with a patch, the SMC will transfer that update to the NGFW Engine and the Engine will also verify the signature on the update (even though the SMC has already verified the update). The Engine will use that update package (which is 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 evaluator can make the patch available to the SMC two different ways, either by saving the patch to an administrator provided USB thumbdrive 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).

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

- NDcPP22e: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.
- VPNGW12: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.
- NDcPP22e: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.
- NDcPP22e: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 on a RW partition on the NGFW Engine. Since the NGFW Engine does not support an interface for local administration, this data is not accessible once stored in the partition.
- NDcPP22e: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 NGFW Engine is updated by the SMC Management Server or alternatively from an administrator configured NTP server.
- NDcPP22e/VPNGW12:FPT_TST_EXT.1/VPNGW12:FPT_TST_EXT.3: The TOE components verify cryptographic algorithms and checksums of TOE binaries upon startup as described above.
- NDcPP22e/VPNGW12:FPT_TUD_EXT.1: The administrator can query the current software versions for the SMC software on the SMC component and for the NGFW Engine software on the SMC Management Server. Administrators can obtain TOE patches from Forcepoint or (in the case of NGFW Engine patches) by configuring the SMC Management Server to automatically download NGFW Engine patches. Administrators must initiate the installation of patches to the NGFW Engine and to the 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.

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:

- NDcPP22e: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.

- VPNGW12:FTA_SSL.3/VPN: The TOE can also terminate VPN client sessions that have been inactive for the administrator defined time interval.
- NDcPP22e: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.
- NDcPP22e:FTA_SSL_EXT.1: The only local interactive sessions are those offered by the Virtual SMC Appliance providing a command line interface.
- NDcPP22e:FTA_TAB.1: A Banner is displayed on both interfaces offered by the Virtual SMC Appliance (i.e., the GUI and local console). The NGFW Engine does not offer a direct network interface.
- VPNGW12:FTA_TSE.1: The TOE can deny VPN client sessions based upon location (as determined by IP address), time and day.
- VPNGW12: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 and 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:

- NDcPP22e: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.
- VPNGW12:FTP_ITC.1/VPN: The TOE's Engines can also protect traffic with remote IPsec peers (either site-to-site or VPN clients).
- NDcPP22e:FTP_TRP.1/Admin: The SMC Management Server's Web UI accepts only TLSv1.2 connections for management operations and
- NDcPP22e: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 NDcPP22e. The following table presents the required mapping.

Requirement	Distributed TOE SFR Allocation	Distributed TOE Audit Event Allocation
NDcPP22e/STFFW14e:FAU_GEN.1	All	All
NDcPP22e:FAU_GEN.2	All	None
VPNGW12:FAU_GEN.1/VPN	Engine(s)	Engine(s)
NDcPP22e:FAU_GEN_EXT.1	All	All
NDcPP22e:FAU_STG_EXT.1	All	None
NDcPP22e:FAU_STG_EXT.4	All	None
NDcPP22e:FAU_STG_EXT.5	All	None
NDcPP22e:FCO_CPC_EXT.1	All	All
NDcPP22e:FCS_CKM.1	All	None
VPNGW12:FCS_CKM.1/IKE	Engine(s)	Engine(s)
NDcPP22e:FCS_CKM.2	All	None
NDcPP22e:FCS_CKM.4	All	None
NDcPP22e:FCS_COP.1/DataEncryption	All	None
VPNGW12:FCS_COP.1/DataEncryption	Engine(s)	Engine(s)
NDcPP22e:FCS_COP.1/SigGen	All	None
NDcPP22e:FCS_COP.1/Hash	All	None
NDcPP22e:FCS_COP.1/KeyedHash	All	None
NDcPP22e:FCS_HTTPS_EXT.1	SMC	SMC
NDcPP22e:FCS_NTP_EXT.1	All	All
NDcPP22e:FCS_RBG_EXT.1	All	None
NDcPP22e:FCS_TLSC_EXT.1	All	All
NDcPP22e:FCS_TLSC_EXT.2	All	All
NDcPP22e:FCS_TLSS_EXT.1	All	All
NDcPP22e:FCS_TLSS_EXT.2	All	All
STFFW14e:FDP_RIP.2	All	None
STFFW14e:FFW_RUL_EXT.1	Engine(s)	Engine(s)
NDcPP22e:FFW_RUL_EXT.2	Engine(s)	None
NDcPP22e:FIA_AFL.1	SMC	SMC
NDcPP22e:FIA_PMG_EXT.1	SMC	None
NDcPP22e:FIA_UAU.7	SMC	None
NDcPP22e:FIA_UAU_EXT.2	SMC	SMC
NDcPP22e:FIA_UIA_EXT.1	SMC	SMC
NDcPP22e:FIA_X509_EXT.1/ITT	All	All
NDcPP22e:FIA_X509_EXT.1/Rev	All	All
VPNGW12:FIA_X509_EXT.1/Rev	Engine(s)	Engine(s)
NDcPP22e:FIA_X509_EXT.2	All	None
VPNGW12:FIA_X509_EXT.2	Engine(s)	Engine(s)
NDcPP22e:FIA_X509_EXT.3	SMC	None
VPNGW12:FIA_X509_EXT.3	Engine(s)	Engine(s)
NDcPP22e:FMT_MOF.1/Functions	All	SMC
NDcPP22e:FMT_MOF.1/ManualUpdate	All	All
NDcPP22e:FMT_MTD.1/CoreData	All	SMC
NDcPP22e:FMT_MOF.1/CryptoKeys	All	All

Requirement	Distributed TOE SFR Allocation	Distributed TOE Audit Event Allocation
VPNGW12:FMT_MOF.1/CryptoKeys	All	SMC
NDcPP22e:FMT_SMF.1	SMC	None
STFFW14e:FMT_SMF.1/FFW	SMC	None
VPNGW12:FMT_SMF.1/VPN	SMC	None
NDcPP22e:FMT_SMR.2	SMC	None
VPNGW12:FPF_RUL_EXT.1	SMC	None
NDcPP22e:FPT_APW_EXT.1	SMC	None
VPNGW12:FPT_FLS.1/SelfTest	Engines	All
NDcPP22e:FPT_ITT.1	All	All
NDcPP22e:FPT_SKP_EXT.1	All	None
NDcPP22e:FPT_STM_EXT.1	All	All
NDcPP22e:FPT_TST_EXT.1	All	None
VPNGW12:FPT_TST_EXT.1	Engine(s)	Engine(s)
VPNGW12:FPT_TST_EXT.3	Engine(s)	Engine(s)
NDcPP22e:FPT_TUD_EXT.1	All	All
VPNGW12:FPT_TUD_EXT.1	All	All
NDcPP22e:FTA_SSL.3	SMC	SMC
NDcPP22e:FTA_SSL.4	SMC	SMC
NDcPP22e:FTA_SSL_EXT.1	SMC	SMC
NDcPP22e:FTA_TAB.1	SMC	None
VPNGW12:FTA_TSE.1	Engine(s)	Engine(s)
VPNGW12:FTA_VCM_EXT.1	Engine(s)	Engine(s)
NDcPP22e:FTP_ITC.1	SMC	SMC
VPNGW12:FTP_ITC.1/VPN	Engine(s)	Engine(s)
NDcPP22e:FTP_TRP.1/Admin	SMC	SMC
NDcPP22e:FTP_TRP.1/Join	All	All