FortiGate/FortiOS 7.2

Security Target

Version 1.5

October 2025

Document prepared by

www.lightshipsec.com

Document History

Version	Date	Author	Description
1.0	14 Mar 2025	G. NICKEL	Release for Check In
1.1	02 May 2025	G. NICKEL	Address ECR Comments
1.2	29 July 2025	G. NICKEL	Update CAVP, Address OR
1.3	15 Aug 2025	G. NICKEL	Address OR, Update TD's
1.4	01 Oct 2025	G. NICKEL	Address ECR comments
1.5	06 Oct 2025	G. NICKEL	Address ECR comments

Table of Contents

List of Tables

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the Fortinet FortiGate/FortiOS 7.2 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 FortiGate next-generation firewall (NGFW) appliances running FortiOS software are designed to provide high performance, multilayered security functionality as described in section 2.3 and allows for granular visibility and protection of enterprise network traffic.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	FortiGate/FortiOS 7.2 Version 7.2.8 (FIPS-CC-72-5, b9663)
Security Target	FortiGate/FortiOS 7.2 Security Target, v1.5

1.3 Conformance Claims

3 This ST supports the following conformance claims:

a) CC version 3.1 revision 5

b) CC Part 2 extended

c) CC Part 3 conformant

d) PP-Configuration for Network Devices, Stateful Traffic Filter Firewalls, and Virtual Private Network Gateways, Version 2.0
(CFG_NDcPP-FW-VPNGW_V2.0)
This PP-Configuration includes the following components:

i) Base-PP: collaborative Protection Profile for Network Devices, Version 3.0e (CPP_ND_V3.0E)

ii) PP-Module: PP-Module for Stateful Traffic Filter Firewalls, Version 1.4e (MOD_CPP_FW_V1.4E)

iii) PP-Module: PP-Module for VPN Gateways, Version 1.3 (MOD_VPNGW_V1.3)

e) Functional Package for Secure Shell (SSH), Version 1.0 (PKG_SSH_V1.0)

f) NIAP Technical Decisions per Table 2

Table 2: NIAP Technical Decisions

TD #	Name	Source	Applicability Rationale
TD0545	NIT Technical Decision for Conflicting FW rules cannot be configured (extension of Rfl#201837)	MOD_CPP_FW_V1.4E	Applicable

TD #	Name	Source	Applicability Rationale
TD0551	NIT Technical Decision for Incomplete Mappings of OEs in FW Module v1.4+Errata	MOD_CPP_FW_V1.4E	Applicable
TD0682	Addressing Ambiguity in FCS_SSHS_EXT.1 Tests	PKG_SSH_V1.0	Applicable
TD0695	Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package.	PKG_SSH_V1.0	Applicable
TD0732	FCS_SSHS_EXT.1.3 Test 2 Update	PKG_SSH_V1.0	Applicable
TD0777	Clarification to Selections for Auditable Events for FCS_SSH_EXT.1	PKG_SSH_V1.0	Applicable
TD0781	Correction to FIA_PSK_EXT.3 EA for MOD_VPNGW_v1.3	MOD_VPNGW_V1.3	Not Applicable – The TOE does not claim FIA_PSK_EXT.3
TD0811	Correction to Referenced SFR in FIA_PSK_EXT.3 Test	MOD_VPNGW_V1.3	Not Applicable – The TOE does not claim FIA_PSK_EXT.3
TD0824	Aligning MOD_VPNGW 1.3 with NDcPP 3.0E	MOD_VPNGW_V1.3	Applicable
TD0827	Aligning MOD_CPP_FW_v1.4E with CPP_ND_V3.0E	MOD_CPP_FW_V1.4E	Applicable
TD0836	NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1	CPP_ND_V3.0E	Applicable
TD0838	PPK Configurability in FIA_PSK_EXT.1.1	MOD_VPNGW_V1.3	Applicable
TD0868	NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8	CPP_ND_V3.0E	Applicable
TD0879	NIT Decision: Correction of Chapter Headings in CPP_ND_V3.0E	CPP_ND_V3.0E	Applicable
TD0880	NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1	CPP_ND_V3.0E	Applicable
TD0886	Clarification to FAU_STG_EXT.1 Test 6	CPP_ND_V3.0E	Applicable
TD0899	NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2	CPP_ND_V3.0E	Applicable

TD #	Name	Source	Applicability Rationale
TD0900	NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3	CPP_ND_V3.0E	Applicable
TD0909	Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0	PKG_SSH_V1.0	Applicable
TD0921	NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment	CPP_ND_V3.0E	Applicable
TD0923	NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2	CPP_ND_V3.0E	Applicable
TD0924	NIT Technical Decision: FFW_RUL_EXT.1.2 Expected Rule Granularity Level	MOD_CPP_FW_V1.4E	Applicable
TD0944	Adding FIPS 186-5 in MOD_VPNGW_V1.3	MOD_VPNGW_V1.3	Applicable

1.4 Terminology

Table 3: Terminology

Term	Definition
BGP	Border Gateway Protocol
CC	Common Criteria
CLI	Command Line Interface
cPP	Collaborative Protection Profile
FW	Firewall
FortiGate	Fortinet NGFW hardware appliance(s)
FortiOS	Fortinet NGFW operating system
GUI	Graphical User Interface
NDcPP	collaborative Protection Profile for Network Devices
NGFW	Next-Generation Firewall
NTP	Network Time Protocol
OSPF	Open Shortest Path First
PP	Protection Profile

Term	Definition
RIP	Routing Information Protocol
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
VDOM	Virtual Domain
VPN	Virtual Private Network

2 TOE Description

2.1 Type

4 The TOE is a firewall with next-generation firewall (NGFW) capabilities including secure Virtual Private Networks (VPN).

2.2 Usage

2.2.1 Deployment

5 As shown in Figure 1, the TOE (enclosed in red) is typically deployed as a gateway between two networks, such as an internal office network and the internet.

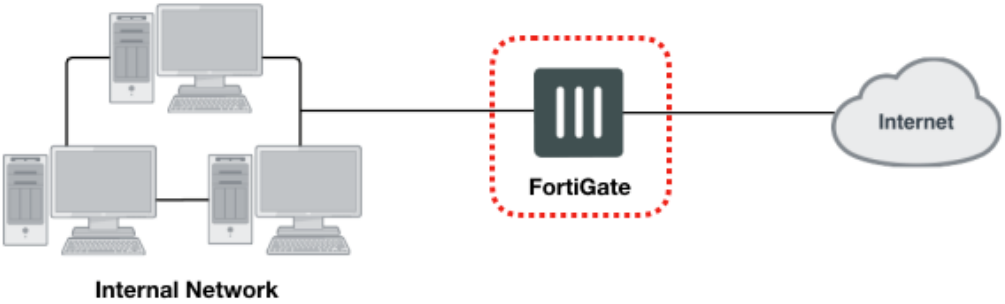

Figure 1: Example TOE deployment

2.2.2 Interfaces

6 The TOE interfaces are shown in Figure 2.

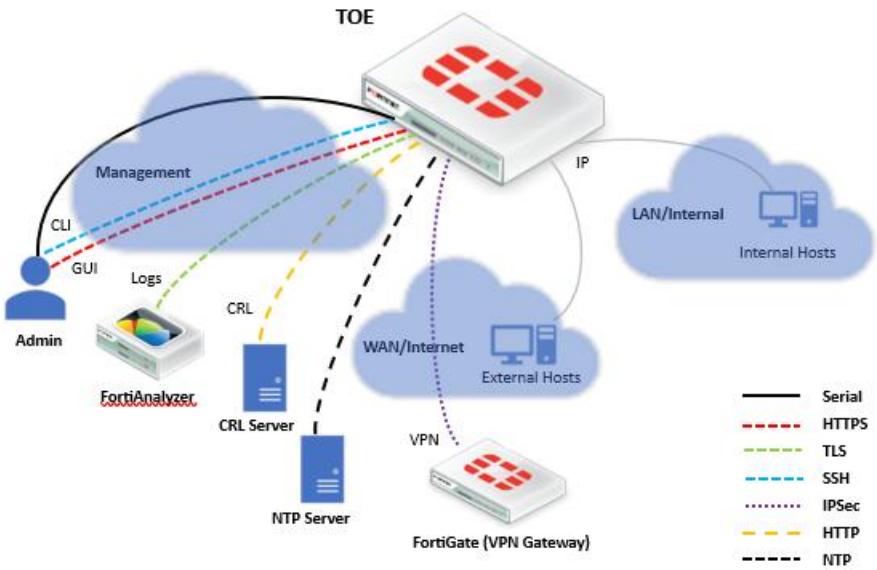

Figure 2: TOE interfaces

7 The logical TOE interfaces are as follows:

a) **CLI.** Administrative CLI via direct serial connection or SSH.

b) **GUI.** Administrative web GUI via HTTPS.

c) **Remote Logging.** Forwarding of TOE audit events to a remote audit server, which is a Fortinet FortiAnalyzer, via TLS.

d) **CRL.** Certificate revocation communication via HTTP.

e) **NTP Server.** Time synchronization via NTP.

f) **VPN Gateway.** VPN connections via IPsec.

g) **WAN/Internet.** External IP interface.

h) **LAN/Internal.** Internal IP interface.

8 **Note:** FortiAnalyzer is the only remote audit server supported for this evaluation because it supports a TLS channel.

2.3 Logical Scope / Security Functions

9 The TOE provides the following security functions:

a) **Security Audit.** The TOE generates logs for auditable events. These logs can be stored locally in protected storage and/or exported to an external audit server via a secure channel.

b) **Cryptographic Support.** The TOE implements a variety of key generation and cryptographic methods to provide protection of data both in transit and at rest within the TOE. In the evaluated configuration, the TOE is in FIPS mode to support the cryptographic functionality. The TOE implements cryptographic protocols such as SSH, TLS, HTTPS, and IPsec.

c) **User Data Protection.** The TOE ensures that residual information and other data cannot be recovered once the associated resource has been deallocated. Data is removed through zeroization.

d) **Identification and Authentication.** The TOE implements mechanisms to ensure that users are both identified and authenticated before any access to TOE functionality or TSF data is granted. Remote login attempts are limited to an administrator-configured threshold, after which the user must wait for a defined period of time before login attempts can be made. It provides the ability to both assign attributes (user names, passwords and roles) and to authenticate users against these attributes. The TOE also provides X.509 certificate validation for its TLS and IPsec connections.

e) **Security Management.** The TOE provides a suite of management functionality, allowing for full configuration of the TOE by an authorized administrator.

f) **Protection of the TSF.** The TOE implements a number of protection mechanisms (including authentication requirements, self-tests and trusted update) to ensure the protection of the TOE and all TSF data. The TOE supports the use of NTP or its own manually configurable time source free from outside interference for the purpose of generating logs and executing reliable time sensitive operations.

g) **TOE Access.** The TOE provides session management functions for local and remote administrative sessions. Administrative sessions have a defined lifetime for both local and remote sessions, users connecting to the TOE will be presented with a warning and consent banner prior to authentication.

h) **Trusted Path/Channels.** The TOE provides secure channels between itself and local/remote administrators and other devices to ensure data security during transit.

i) **Stateful Traffic and Packet Filtering.** The TOE allows for the configuration and enforcement of stateful packet filtering/firewall rules on all traffic traversing the TOE.

2.3.1 Functions not included in the TOE Evaluation

10 The FortiGate appliances are capable of a variety of functions and configurations which are not covered by the claimed PP-Configuration.

11 The following features have not been examined as part of this evaluation:

a) High-Availability;

b) FortiExplorer client;

c) FortiGuard Anti-spam, Firmware, Anti-Virus, Content Filtering, Web Filtering, EndPoint Control, and FortiSandbox services;

d) Use of syslog;

e) FortiToken and FortiSSO Authentication;

f) Stream Control Transmission Protocol (SCTP), BGP, RIP, and OSPF protocols;

g) Usage of the boot-time configuration menu to upgrade the TOE;

h) Policy-based VPN;

i) SSL VPN;

j) Logging to FortiCloud;

k) External Threat Feeds;

l) Explicit Web Proxy with Form-Based Authentication;

m) REST API;

n) Traffic Shaping;

o) SMTP;

p) SNMP;

q) LDAP;

r) Windows AD;

s) RADIUS;

t) USB Interface

u) Diagnostics interface;

v) DHCP, DDNS, or DNS;

w) Traffic offloading to FortiASIC NPx network processors.

x) IPS

y) HTTP GUI

z) Telnet (TOE acting as client or server)

aa) TFTP (TOE acting as client)

bb) Precision Time Protocol (PTP)

2.4 Physical Scope

12 The physical boundary of the TOE includes the FortiGate hardware models shown in Table 4 running FortiOS software identified in Table 1. The TOE is shipped to the customer via commercial courier.

13 Fortinet Virtual Domains (VDOMs) are supported by default on the appliances listed in Table 4. Multi-VDOM mode with Independent VDOM configuration was included in the evaluated configuration.

Table 4: TOE Hardware Models

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy	ESV	ASIC CAVP	FortiOS FIPS CAVP	FortiOS SSL CAVP
FG-40F	Fortinet SoC4	ARMv8	2 GB	4GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FG-40F-3G4G	Fortinet SoC4	ARMv8	2 GB	4GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FWF-40F	Fortinet SoC4	ARMv8	2 GB	4GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FWF-40F-3G4G	Fortinet SoC4	ARMv8	2 GB	4GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FG-60E	Fortinet SoC3	ARMv7-A	2 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-60E-PoE	Fortinet SoC3	ARMv7-A	2 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-60E-DSL	Fortinet SoC3	ARMv7-A	2 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FWF-60E	Fortinet SoC3	ARMv7-A	2 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FWF-60E-DSL	Fortinet SoC3	ARMv7-A	2 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FWF-60E-DSLJ	Fortinet SoC3	ARMv7-A	2 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-60F	Fortinet SoC4	ARMv8	2 GB	8GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy	ESV	ASIC CAVP	FortiOS FIPS CAVP	FortiOS SSL CAVP
FGR-60F-3G4G	Fortinet SoC4	ARMv8	2 GB	8GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FGR-60F	Fortinet SoC4	ARMv8	2 GB	8GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FWF-60F	Fortinet SoC4	ARMv8	2 GB	8GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FG-61E	Fortinet SoC3	ARMv7-A	2 GB	8GB	128GB	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FWF-61E	Fortinet SoC3	ARMv7-A	2 GB	8GB	128GB	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-61F	Fortinet SoC4	ARMv8	2 GB	8GB	128GB	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FGR-70F	Fortinet SoC4	ARMv8	4 GB	8GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FGR-70F-3G4G	Fortinet SoC4	ARMv8	4 GB	8GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FG-70F	Fortinet SoC4	ARMv8	4 GB	8GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FG-71F	Fortinet SoC4	ARMv8	4 GB	8GB	128GB	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FG-80E	Fortinet SoC3	ARMv7-A	2 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-80E-PoE	Fortinet SoC3	ARMv7-A	2 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy	ESV	ASIC CAVP	FortiOS FIPS CAVP	FortiOS SSL CAVP
FG-80F	Fortinet SoC4	ARMv8	4 GB	8GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FG-80F-PoE	Fortinet SoC4	ARMv8	4 GB	8GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FWF-80F-2R	Fortinet SoC4	ARMv8	4 GB	8GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FG-81E	Fortinet SoC3	ARMv7-A	2 GB	8GB	128GB	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-81E-PoE	Fortinet SoC3	ARMv7-A	2 GB	8GB	128GB	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-81F	Fortinet SoC4	ARMv8	4 GB	8GB	128GB	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FG-81F-PoE	Fortinet SoC4	ARMv8	4 GB	8GB	128GB	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FWF-81F-2R	Fortinet SoC4	ARMv8	4 GB	8GB	128GB	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FWF-81F-2R-3G4G-PoE	Fortinet SoC4	ARMv8	4 GB	8GB	128GB	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FWF-81F-2R-PoE	Fortinet SoC4	ARMv8	4 GB	8GB	128GB	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-90E	Fortinet SoC3	ARMv7-A	2 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-91E	Fortinet SoC3	ARMv7-A	2 GB	8GB	128GB	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy	ESV	ASIC CAVP	FortiOS FIPS CAVP	FortiOS SSL CAVP
FG-100E	Fortinet SoC3	ARMv7-A	4 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-100EF	Fortinet SoC3	ARMv7-A	4 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-100F	Fortinet SoC4	ARMv8	4 GB	8GB	N/A	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FG-101E	Fortinet SoC3	ARMv7-A	4 GB	8GB	480GB	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-101F	Fortinet SoC4	ARMv8	4 GB	8GB	480GB	CP9 XLite	JitterEnt	E139	A2242	A6641	A6643
FG-140E	Fortinet SoC3	ARMv7-A	4 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-140E-PoE	Fortinet SoC3	ARMv7-A	4 GB	8GB	N/A	CP9 Lite	JitterEnt	E139	A2241	A6641	A6643
FG-200E	Intel Celeron G1820	Haswell	4GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-200F	Intel Xeon D-1627	Hewitt Lake	8GB	30GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-201E	Intel Celeron G1820	Haswell	4GB	16GB	480GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-201F	Intel Xeon D-1627	Hewitt Lake	8GB	30GB	480GB	CP9	JitterEnt	E139	A2240	A6641	A6643

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy	ESV	ASIC CAVP	FortiOS FIPS CAVP	FortiOS SSL CAVP
FG-300E	Intel Core i5-6500	SkyLake	8GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-301E	Intel Core i5-6500	SkyLake	8GB	16GB	480GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-400E	Intel Core i5-8500	Coffee Lake	8GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-400F	Intel Xeon E-2336	Cypress Cove (Rocket Lake)	16GB	30GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-401E	Intel Core i5-8500	Coffee Lake	8GB	16GB	480GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-401E-DC	Intel Core i5-8500	Coffee Lake	8GB	16GB	480GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-401F	Intel Xeon E-2336	Cypress Cove (Rocket Lake)	16GB	30GB	960GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-500E	Intel Core i7-6700	SkyLake	16GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-501E	Intel Core i7-6700	SkyLake	16GB	16GB	480GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-600E	Intel Core i7-8700	Coffee Lake	16 GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-600F	Intel Xeon E-2386G	Cypress Cove	16 GB	30GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy	ESV	ASIC CAVP	FortiOS FIPS CAVP	FortiOS SSL CAVP
		(Rocket Lake)									
FG-601E	Intel Core i7-8700	Coffee Lake	16 GB	16GB	480GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-601F	Intel Xeon E-2386G	Cypress Cove (Rocket Lake)	16 GB	30GB	480GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-900G	AMD RYZEN 5950E	Zen 3 (Vermeer)	32 GB	30GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-901G	AMD RYZEN 5950E	Zen 3 (Vermeer)	32 GB	30GB	960GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-1000F	Intel Xeon E-2388G	Cypress Cove (Rocket Lake)	16GB	30GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-1001F	Intel Xeon E-2388G	Cypress Cove (Rocket Lake)	16GB	30GB	960GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-1100E	Intel Xeon E-2186G	Coffee Lake	16 GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-1100E-DC	Intel Xeon E-2186G	Coffee Lake	16 GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-1101E	Intel Xeon E-2186G	Coffee Lake	16 GB	16GB	960GB	CP9	JitterEnt	E139	A2240	A6641	A6643

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy	ESV	ASIC CAVP	FortiOS FIPS CAVP	FortiOS SSL CAVP
FG-1101E-DC	Intel Xeon E-2186G	Coffee Lake	16 GB	16GB	960GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-1800F	Intel Xeon W-3223	Cascade Lake	24GB	30GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-1800F-DC	Intel Xeon W-3223	Cascade Lake	24GB	30GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-1801F	Intel Xeon W-3223	Cascade Lake	24GB	30GB	2TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-1801F-DC	Intel Xeon W-3223	Cascade Lake	24GB	30GB	2TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-2000E	Intel Xeon E5-1660v4	Broadwell	32 GB	16GB	480GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-2200E	Intel Xeon Gold 6126	SkyLake	24 GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-2201E	Intel Xeon Gold 6126	SkyLake	24 GB	16GB	2TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-2500E	Intel Xeon E5-1650v3	Haswell	32 GB	16GB	480GB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-2600F	Intel Xeon Gold 6208U	Cascade Lake	48 GB	30 GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy	ESV	ASIC CAVP	FortiOS FIPS CAVP	FortiOS SSL CAVP
FG-2601F	Intel Xeon Gold 6208U	Cascade Lake	48 GB	30 GB	2 TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-2601F-DC	Intel Xeon Gold 6208U	Cascade Lake	48 GB	30 GB	2 TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3000F	AMD EPYC 7502P	Zen 2 (Rome)	128 GB	30GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3000F-DC	AMD EPYC 7502P	Zen 2 (Rome)	128 GB	30GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3001F	AMD EPYC 7502P	Zen 2 (Rome)	128 GB	30GB	2TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3001F-DC	AMD EPYC 7502P	Zen 2 (Rome)	128 GB	30GB	2TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3200F	Intel Xeon Gold 6348	Ice Lake	128GB	30GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3201F	Intel Xeon Gold 6348	Ice Lake	128GB	30GB	2 TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3300E	Intel Xeon Gold 5118	SkyLake	96 GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy	ESV	ASIC CAVP	FortiOS FIPS CAVP	FortiOS SSL CAVP
FG-3301E	Intel Xeon Gold 5118	SkyLake	96 GB	16GB	2TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3400E	Intel Xeon Gold 6130	SkyLake	96 GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3400E-DC	Intel Xeon Gold 6130	SkyLake	96 GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3401E	Intel Xeon Gold 6130	SkyLake	96 GB	16GB	2TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3401E-DC	Intel Xeon Gold 6130	SkyLake	96 GB	16GB	2TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3500F	AMD EPYC 7542	Zen 2 (Rome)	256 GB	30GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3501F	AMD EPYC 7542	Zen 2 (Rome)	256 GB	30GB	4TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3600E	Intel Xeon Gold 6152	SkyLake	96 GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3600E-DC	Intel Xeon Gold 6152	SkyLake	96 GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy	ESV	ASIC CAVP	FortiOS FIPS CAVP	FortiOS SSL CAVP
FG-3601E	Intel Xeon Gold 6152	SkyLake	96 GB	16GB	2TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3700F	Intel Xeon Gold 6348	Ice Lake	256GB	30GB	4TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3701F	Intel Xeon Gold 6348	Ice Lake	256GB	30GB	4TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3960E	Intel Xeon E5-2650V4	Broadwell	256GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3960E-DC	Intel Xeon E5-2650V4	Broadwell	256GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3980E	Intel Xeon E5-2680V4	Broadwell	256GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-3980E-DC	Intel Xeon E5-2680V4	Broadwell	256GB	16GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-4200F	Intel Xeon Gold 6248	Cascade Lake	384 GB	30 GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-4200F-DC	Intel Xeon Gold 6248	Cascade Lake	384 GB	30 GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy	ESV	ASIC CAVP	FortiOS FIPS CAVP	FortiOS SSL CAVP
FG-4201F	Intel Xeon Gold 6248	Cascade Lake	384 GB	30 GB	4 TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-4201F-DC	Intel Xeon Gold 6248	Cascade Lake	384 GB	30 GB	4 TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-4400F	Intel Xeon Gold 6248	Cascade Lake	384 GB	30 GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-4400F-DC	Intel Xeon Gold 6248	Cascade Lake	384 GB	30 GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-4401F	Intel Xeon Gold 6248	Cascade Lake	384 GB	30 GB	4 TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-4401F-DC	Intel Xeon Gold 6248	Cascade Lake	384 GB	30 GB	4 TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-4800F	Intel Xeon Gold 6348	Ice Lake	512 GB	30GB	N/A	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-4801F	Intel Xeon Gold 6348	Ice Lake	512 GB	30GB	4 TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-5001E1	Intel Xeon E5-2690v4	Broadwell	64GB	16GB	480 GB	CP9	JitterEnt	E139	A2240	A6641	A6643

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy	ESV	ASIC CAVP	FortiOS FIPS CAVP	FortiOS SSL CAVP
FG-6001F	Intel Xeon D-1567	Broadwell	64GB	30GB	2 TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-6300F	Intel Xeon D-1567	Broadwell	192GB	16GB	2 TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-6301F	Intel Xeon D-1567	Broadwell	192GB	16GB	2 TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-6500F	Intel Xeon D-1567	Broadwell	320GB	16GB	2 TB	CP9	JitterEnt	E139	A2240	A6641	A6643
FG-6501F	Intel Xeon D-1567	Broadwell	320GB	16GB	2 TB	CP9	JitterEnt	E139	A2240	A6641	A6643

2.4.1 Guidance Documents

14 The TOE includes the following guidance documents (PDF):

a) FIPS 140-3 and NDcPP Technote for FortiOS 7.2 and FortiGate NGFW Appliances, 01-728-1075780-20251006, October 6, 2025

b) FortiOS 7.2.8 Administration Guide, 01-728-791905-20240807, August 7 2024

c) FortiOS 7.2.8 CLI Reference, 01-728-791773-20240314, March 14 2024

d) FortiOS 7.2.8 Log Reference, 01-728-791443-20240314, March 14 2024

e) FortiOS 7.2.8 Hardware Acceleration Guide, 01-729-538746-20240314, March 14 2024

f) NDcPP Logging Addendum for FortiOS 7.2 and FortiGate NGFW Appliances, 01-728-1172697-20250620, June 20, 2025

g) FortiOS 6.4.0 Parallel Path Processing, 01-640-619132-20210125

2.4.2 Non-TOE Components

15 The TOE operates with the following components in the environment:

a) **Admin's Workstation.** The TOE makes use of a separate workstation for administrative purposes.

b) **Audit Server.** The TOE makes use of a FortiAnalyzer for remote logging.

c) **VPN Endpoints.** The TOE supports FortiGate VPN endpoints.

d) **CRL Web Server.** Web server capable of serving up CRLs over HTTP.

e) **NTP Server.** The TOE makes use of an NTP server to provide reliable and synchronized time information.

3 Security Problem Definition

16 The Security Problem Definition is reproduced from the claimed Protection Profiles.

3.1 Threats

Table 5: Threats (CPP_ND)

Identifier	Description
T.UNAUTHORIZED_ ADMINISTRATOR_ ACCESS	Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_ CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow

Identifier	Description
	attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_ COMMUNICATION_ CHANNELS	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.
T.WEAK_ AUTHENTICATION_ ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised.
T.UPDATE_ COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_ FUNCTIONALITY_ COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker. Threat agents may also be able to take advantage of weak administrative passwords to gain privileged access to the device.
T.SECURITY_ FUNCTIONALITY_ FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

Table 6: Threats (MOD_CPP_FW)

Identifier	Description
T.NETWORK_DISCLOSURE	An attacker may attempt to “map” a subnet to determine the machines that reside on the network, and obtaining the IP addresses of machines, as well as the services (ports) those machines are offering. This information could be used to mount attacks to those machines via the services that are exported.
T.NETWORK_ACCESS	With knowledge of the services that are exported by machines on a subnet, an attacker may attempt to exploit those services by mounting attacks against those services.
T.NETWORK_MISUSE	An attacker may attempt to use services that are exported by machines in a way that is unintended by a site’s security policies. For example, an attacker might be able to use a service to “anonymize” the attacker’s machine as they mount attacks against others.
T.MALICIOUS_TRAFFIC	An attacker may attempt to send malformed packets to a machine in hopes of causing the network stack or services listening on UDP/TCP ports of the target machine to crash.

Table 7: Threats (MOD_VPNGW)

Identifier	Description
T.DATA_INTEGRITY	Devices on a protected network may be exposed to threats presented by devices located outside the protected network that may attempt to modify the data without authorization. If known malicious external devices are able to communicate with devices on the protected network or if devices on the protected network can communicate with those external devices then the data contained in the communications may be susceptible to a loss of integrity.
T.NETWORK_ACCESS	Devices located outside the protected network may seek to exercise services located on the protected network that are intended to only be accessed from inside the protected network or only accessed by entities using an authenticated path into the protected network. Devices located outside the protected network may, likewise, offer services that are inappropriate for access from within the protected network. From an ingress perspective, VPN gateways can be configured so that only those network servers intended for external consumption by entities operating on a trusted network (e.g., machines operating on a network where the peer VPN gateways are supporting the connection) are accessible and only via the intended ports. This serves to mitigate the potential for network entities outside a protected network to access network servers or services intended only for consumption or access inside a protected network. From an egress perspective, VPN gateways can be configured so that only specific external services (e.g., based on destination port) can be accessed from within a protected network, or moreover are accessed via an encrypted channel. For example, access to external mail services can be blocked to enforce corporate policies against

Identifier	Description
	accessing uncontrolled e-mail servers, or, that access to the mail server must be done over an encrypted link.
T.NETWORK_DISCLOSURE	Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to conduct unauthorized activities. If known malicious external devices are able to communicate with devices on the protected network, or if devices on the protected network can establish communications with those external devices (e.g., as a result of a phishing episode or by inadvertent responses to email messages), then those internal devices may be susceptible to the unauthorized disclosure of information. From an infiltration perspective, VPN gateways serve not only to limit access to only specific destination network addresses and ports within a protected network, but whether network traffic will be encrypted or transmitted in plaintext. With these limits, general network port scanning can be prevented from reaching protected networks or machines, and access to information on a protected network can be limited to that obtainable from specifically configured ports on identified network nodes (e.g., web pages from a designated corporate web server). Additionally, access can be limited to only specific source addresses and ports so that specific networks or network nodes can be blocked from accessing a protected network thereby further limiting the potential disclosure of information. From an exfiltration perspective, VPN gateways serve to limit how network nodes operating on a protected network can connect to and communicate with other networks limiting how and where they can disseminate information. Specific external networks can be blocked altogether or egress could be limited to specific addresses and/or ports. Alternately, egress options available to network nodes on a protected network can be carefully managed in order to, for example, ensure that outgoing connections are encrypted to further mitigate inappropriate disclosure of data through packet sniffing.
T.NETWORK_MISUSE	Devices located outside the protected network, while permitted to access particular public services offered inside the protected network, may attempt to conduct inappropriate activities while communicating with those allowed public services. Certain services offered from within a protected network may also represent a risk when accessed from outside the protected network. From an ingress perspective, it is generally assumed that entities operating on external networks are not bound by the use policies for a given protected network. Nonetheless, VPN gateways can log policy violations that might indicate violation of publicized usage statements for publicly available services. From an egress perspective, VPN gateways can be configured to help enforce and monitor protected network use policies. As explained in the other threats, a VPN gateway can serve to limit dissemination of data, access to external servers, and even disruption of services – all of these could be related to the use policies of a protected network and as such are subject in some regards to enforcement. Additionally, VPN gateways can be configured to log

Identifier	Description
	network usages that cross between protected and external networks and as a result can serve to identify potential usage policy violations.
T.REPLAY_ATTACK	If an unauthorized individual successfully gains access to the system, the adversary may have the opportunity to conduct a “replay” attack. This method of attack allows the individual to capture packets traversing throughout the network and send the packets at a later time, possibly unknown by the intended receiver. Traffic is subject to replay if it meets the following conditions: • Cleartext: an attacker with the ability to view unencrypted traffic can identify an appropriate segment of the communications to replay as well in order to cause the desired outcome. • No integrity: alongside cleartext traffic, an attacker can make arbitrary modifications to captured traffic and replay it to cause the desired outcome if the recipient has no means to detect these.

3.2 Assumptions

Table 8: Assumptions (CPP_ND)

Identifier	Description
A.PHYSICAL_ PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device’s physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.
A.LIMITED_ FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality). If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform.

Identifier	Description
A.NO_THRU_ TRAFFIC_ PROTECTION	A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the NDcPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).
A.TRUSTED_ ADMINISTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
A.REGULAR_ UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_ CREDENTIALS_ SECURE	The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.
A.RESIDUAL_ INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

Table 9: Assumptions (MOD_VPNGW)

Identifier	Description
A.CONNECTIONS	It is assumed that 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.

3.3 Organizational Security Policies

Table 10: Organizational Security Policies (CPP_ND)

Identifier	Description
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which Administrators consent by accessing the TOE.

4 Security Objectives

4.1 Security Objectives for the TOE

Table 11: Security Objectives for the TOE (MOD_CPP_FW)

Identifier	Description
O.RESIDUAL_ INFORMATION	The TOE shall implement measures to ensure that any previous information content of network packets sent through the TOE is made unavailable either upon deallocation of the memory area containing the network packet or upon allocation of a memory area for a newly arriving network packet or both.
O.STATEFUL_TRAFFIC_ FILTERING	The TOE shall perform stateful traffic filtering on network packets that it processes. For this the TOE shall support the definition of stateful traffic filtering rules that allow to permit or drop network packets. The TOE shall support assignment of the stateful traffic filtering rules to each distinct network interface. The TOE shall support the processing of the applicable stateful traffic filtering rules in an administratively defined order. The TOE shall deny the flow of network packets if no matching stateful traffic filtering rule is identified. Depending on the implementation, the TOE might support the stateful traffic filtering of Dynamic Protocols (optional).

Table 12: Security Objectives for the TOE (MOD_VPNGW)

Identifier	Description
O.ADDRESS_FILTERING	To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption or denial of services, and network-based reconnaissance, compliant TOE's will implement packet filtering capability. That capability will restrict the flow of network traffic between protected networks and other attached networks based on network addresses of the network nodes originating (source) and/or receiving (destination) applicable network traffic as well as on established connection information.

Identifier	Description
O.AUTHENTICATION	To further address the issues associated with unauthorized disclosure of information, a compliant TOE's authentication ability (IPSec) will allow a VPN peer to establish VPN connectivity with another VPN peer and ensure that any such connection attempt is both authenticated and authorized. VPN endpoints authenticate each other to ensure they are communicating with an authorized external IT entity.
O.CRYPTOGRAPHIC_ FUNCTIONS	To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption of services, and network-based reconnaissance, compliant TOE's will implement a cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.
O.FAIL_SECURE	There may be instances where the TOE's hardware malfunctions or the integrity of the TOE's software is compromised, the latter being due to malicious or non-malicious intent. To address the concern of the TOE operating outside of its hardware or software specification, the TOE will shut down upon discovery of a problem reported via the self-test mechanism and provide signature-based validation of updates to the TSF.
O.PORT_FILTERING	To further address the issues associated with unauthorized disclosure of information, etc., a compliant TOE's port filtering capability will restrict the flow of network traffic between protected networks and other attached networks based on the originating (source) and/or receiving (destination) port (or service) identified in the network traffic as well as on established connection information.
O.SYSTEM_MONITORING	To address the issues of administrators being able to monitor the operations of the VPN gateway, it is necessary to provide a capability to monitor system activity. Compliant TOEs will implement the ability to log the flow of network traffic. Specifically, the TOE will provide the means for administrators to configure packet filtering rules to 'log' when network traffic is found to match the configured rule. As a result, matching a rule configured to 'log' will result in informative event logs whenever a match occurs. In addition, the establishment of security associations (SAs) is auditable, not only between peer VPN gateways, but also with certification authorities (CAs).
O.TOE_ADMINISTRATION	TOEs will provide the functions necessary for an administrator to configure the packet filtering rules, as well as the cryptographic aspects of the IPsec protocol that are enforced by the TOE.

4.2 Security Objectives for the Environment

Table 13: Security Objectives for the Environment (CPP_ND)

Identifier	Description
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_ PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. 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.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE’s trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_ CREDENTIALS_ SECURE	The Administrator’s credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.RESIDUAL_ INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.

Table 14: Security Objectives for the Environment (MOD_VPNGW)

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

5 Security Requirements

5.1 Conventions

17 This document uses the following font conventions to identify the operations defined by the CC:

a) **Assignment.** Indicated with italicized text.

b) **Refinement.** Indicated with bold text and strikethroughs.

c) **Selection.** Indicated with underlined text.

d) **Assignment within a Selection:** Indicated with italicized and underlined text.

5.2 Extended Components Definition

18 Refer to Annex A of this ST.

5.3 Functional Requirements

Table 15: Summary of SFRs

Requirement	Title	Source
FAU_GEN.1	Audit Data Generation	CPP_ND_V3.0E MOD_CPP_FW_V1.4E PKG_SSH_V1.0
FAU_GEN.1/VPN	Audit Data Generation (VPN Gateway)	MOD_VPNGW_V1.3
FAU_GEN.2	User identity association	CPP_ND_V3.0E
FAU_STG_EXT.1	Protected Audit Event Storage	CPP_ND_V3.0E
FCS_CKM.1	Cryptographic Key Generation	CPP_ND_V3.0E
FCS_CKM.1/IKE	Cryptographic Key Generation (for IKE Peer Authentication)	MOD_VPNGW_V1.3
FCS_CKM.2	Cryptographic Key Establishment	CPP_ND_V3.0E
FCS_CKM.4	Cryptographic Key Destruction	CPP_ND_V3.0E

Requirement	Title	Source
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)	CPP_ND_V3.0E MOD_VPNGW_V1.3
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)	CPP_ND_V3.0E
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)	CPP_ND_V3.0E
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)	CPP_ND_V3.0E
FCS_RBG_EXT.1	Random Bit Generation	CPP_ND_V3.0E
FCS_HTTPS_EXT.1	HTTPS Protocol	CPP_ND_V3.0E
FCS_SSH_EXT.1	SSH Protocol	PKG_SSH_V1.0
FCS_SSHS_EXT.1	SSH Protocol – Server	PKG_SSH_V1.0
FCS_TLSC_EXT.1	TLS Client Protocol	CPP_ND_V3.0E
FCS_TLSC_EXT.2	TLS Client Support for Mutual Authentication	CPP_ND_V3.0E
FCS_TLSS_EXT.1	TLS Server Protocol	CPP_ND_V3.0E
FCS_IPSEC_EXT.1	IPsec Protocol	CPP_ND_V3.0E MOD_VPNGW_V1.3
FCS_NTP_EXT.1	NTP Protocol	CPP_ND_V3.0E
FDP_RIP.2	Full Residual Information Protection	MOD_CPP_FW_V1.4E
FIA_AFL.1	Authentication Failure Handling	CPP_ND_V3.0E
FIA_PMG_EXT.1	Password Management	CPP_ND_V3.0E
FIA_PSK_EXT.1	Pre-Shared Key Composition	MOD_VPNGW_V1.3
FIA_PSK_EXT.2	Generated Pre-Shared Keys	MOD_VPNGW_V1.3
FIA_UAU.7	Protected Authentication Feedback (Refinement)	CPP_ND_V3.0E
FIA_UIA_EXT.1	User Identification and Authentication	CPP_ND_V3.0E
FIA_X509_EXT.1/Rev	X.509 Certificate Validation	CPP_ND_V3.0E MOD_VPNGW_V1.3
FIA_X509_EXT.2	X.509 Certificate Authentication	CPP_ND_V3.0E

Requirement	Title	Source
		MOD_VPNGW_V1.3
FIA_X509_EXT.3	X.509 Certificate Requests	CPP_ND_V3.0E MOD_VPNGW_V1.3
FMT_MOF.1/ManualUpdate	Management of Security Functions Behaviour	CPP_ND_V3.0E
FMT_MOF.1/Functions	Management of Security Functions Behaviour	CPP_ND_V3.0E
FMT_MOF.1/Services	Management of Security Functions Behaviour	CPP_ND_V3.0E
FMT_MTD.1/CoreData	Management of TSF Data	CPP_ND_V3.0E
FMT_MTD.1/CryptoKeys	Management of TSF Data	CPP_ND_V3.0E MOD_VPNGW_V1.3
FMT_SMF.1	Specification of Management Functions	CPP_ND_V3.0E
FMT_SMF.1/FFW	Specification of Management Functions	MOD_CPP_FW_V1.4E
FMT_SMF.1/VPN	Specification of Management Functions	MOD_VPNGW_V1.3
FMT_SMR.2	Restrictions on Security Roles	CPP_ND_V3.0E
FPT_APW_EXT.1	Protection of Administrator Passwords	CPP_ND_V3.0E
FPT_FLS.1/SelfTest	Failure with Preservation of Secure State (Self-Test Failures)	MOD_VPNGW_V1.3
FPT_TST_EXT.1	TSF Testing	CPP_ND_V3.0E MOD_VPNGW_V1.3
FPT_TST_EXT.3	Self-Test with Defined Methods	MOD_VPNGW_V1.3
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)	CPP_ND_V3.0E
FPT_TUD_EXT.1	Trusted Update	CPP_ND_V3.0E MOD_VPNGW_V1.3
FPT_STM_EXT.1	Reliable Time Stamps	CPP_ND_V3.0E
FTA_SSL_EXT.1	TSF-initiated Session Locking	CPP_ND_V3.0E
FTA_SSL.3	TSF-initiated Termination	CPP_ND_V3.0E
FTA_SSL.4	User-initiated Termination	CPP_ND_V3.0E

Requirement	Title	Source
FTA_TAB.1	Default TOE Access Banners	CPP_ND_V3.0E
FTP_ITC.1	Inter-TSF Trusted Channel	CPP_ND_V3.0E
FTP_ITC.1/VPN	Inter-TSF Trusted Channel (VPN Communications)	MOD_VPNGW_V1.3
FTP_TRP.1/Admin	Trusted Path	CPP_ND_V3.0E
FFW_RUL_EXT.1	Stateful Traffic Filtering	MOD_CPP_FW_V1.4E
FPF_RUL_EXT.1	Packet Filtering Rules	MOD_VPNGW_V1.3

5.3.1 Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

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

a) Start-up and shutdown of the audit functions;

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

c) All administrative actions comprising:

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

d) *Specifically defined auditable events listed in* ~~***Table 2***~~ ***Table 16***.

Table 16: SFRs and Auditable Events

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

Requirement	Auditable Events	Additional Audit Record Contents
FAU_STG_EXT.1	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session	Reason for failure
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure
FCS_NTP_EXT.1	• Configuration of a new time server • Removal of configured time server	Identity if new/removed time server
FCS_SSH_EXT.1	[Failure to establish SSH connection]	[Reason for failure and Non-TOE endpoint of attempted connection (IP Address)]
FCS_SSH_EXT.1	[Establishment of SSH connection]	[Non-TOE endpoint of connection (IP Address)]
FCS_SSH_EXT.1	[Termination of SSH connection session]	[Non-TOE endpoint of connection (IP Address)]
FCS_SSH_EXT.1	[Dropping of packet(s) outside defined size limits]	[Packet size]
FCS_SSHS_EXT.1	No events specified	None.
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure.
FCS_TLSC_EXT.2	None	None
FCS_TLSS_EXT.1	Failure to establish a TLS session	Reason for failure

Requirement	Auditable Events	Additional Audit Record Contents
FDP_RIP.2	None	None
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanisms.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
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
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None	None
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Functions	None.	None.
FMT_MOF.1/Services	None.	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMF.1/FFW	All management activities of TSF data (including creation, modification and deletion of firewall rules).	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time – either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session lock.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	• None • None • Reason for failure
FTP_TRP.1/Admin	• Initiation of the trusted path. • Termination of the trusted path. • Failure of the trusted path functions.	• None • None • Reason for failure
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

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 ~~(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 cPP/ST, *information specified in column three of ~~Table 2~~* ***Table 16 & Table 17***.

FAU_GEN.1/VPN Audit Data Generation (VPN Gateway)

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

a) Start-up and shutdown of the audit functions

b) Indication that TSF self-test was completed

c) Failure of self-test

d) All auditable events for the [*not specified*] level of audit; and

e) [*auditable events defined in the Auditable Events for Mandatory Requirements table*].

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

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

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

Table 17: Auditable Events for Mandatory Requirements (MOD_VPNGW)

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1 /VPN	No events specified.	N/A
FCS_CKM.1/IKE	No events specified.	N/A
FMT_SMF.1/VPN	All administrative actions	No additional information.
FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	• Source and destination addresses • Source and destination ports • Transport Layer Protocol
FPT_FLS.1/SelfTest	No events specified.	N/A
FPT_TST_EXT.3	No events specified.	N/A
FTP_ITC.1/VPN	• Initiation of the trusted channel	• No additional information.

Requirement	Auditable Events	Additional Audit Record Contents
	• Termination of the trusted channel • Failure of the trusted channel functions	• No additional information. • Identification of the initiator and target of failed trusted channel establishment attempt
FIA_PSK_EXT.1	No events specified	N/A
FIA_PSK_EXT.2	No events specified	N/A

FAU_GEN.2 User Identity Association

FAU_GEN.2.1 For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

FAU_STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.

FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself. In addition [

- The TOE shall consist of a single standalone component that stores audit data locally].

FAU_STG_EXT.1.3 The TSF shall maintain a [buffer] of audit records in the event that an interruption of communication with the remote audit server occurs.

FAU_STG_EXT.1.4 The TSF shall be able to store [non-persistent] audit records locally with a minimum storage size of *[file of 10mb and a configurable maximum of up to 5% of FortiGate model system memory].*

FAU_STG_EXT.1.5 The TSF shall [overwrite previous audit records according to the following rule: *[delete the oldest stored audit logs]*] when the local storage space for audit data is full.

FAU_STG_EXT.1.6 The TSF shall provide the following mechanisms for administrative access to locally stored audit records [ability to view locally].

5.3.2 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

FCS_CKM.1.1 The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- RSA schemes using cryptographic key sizes of *[2048 bits, 4096 bits*] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3 or FIPS PUB 186-5, “Digital Signature Standard (DSS)”, A.1;
- ECC schemes using ‘NIST curves’ [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4, or FIPS PUB 186-5, “Digital Signature Standard (DSS)”, Appendix A.2, or ISO/IEC 14888-3, “IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms”, Section 6.6.;
- FFC Schemes using ‘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, RFC 7919].

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

Application Note: This SFR has been modified by TD0921.

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

FCS_CKM.1.1/IKE The TSF shall generate **asymmetric** cryptographic keys **used for IKE peer authentication** in accordance with a specified cryptographic key generation algorithm: **[**

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

and [

- no other key generation algorithm**]**

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

Application Note: This SFR has been modified by TD0944.

FCS_CKM.2 Cryptographic Key Establishment

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

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;
- FFC Schemes using “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, groups listed in RFC 7919]

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

FCS_CKM.4 Cryptographic Key Destruction

FCS_CKM.4.1 The TSF shall destroy cryptographic keys in accordance with a specified cryptographic key destruction method:

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

that meets the following: *No Standard*.

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

FCS_COP.1.1/DataEncryption The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in **[CBC, GCM] and [no other]** mode and cryptographic key sizes **[128 bits, 256 bits], and [no other cryptographic key sizes]** 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].**

Application Note: *This SFR has been modified from its definition in the NDcPP to support this PP-Module's IPsec requirements by mandating support for at least one of CBC or GCM modes and at least one of 128-bit or 256-bit key sizes at minimum. Other selections may be made by the ST author but they are not required for conformance to this PP-Module.*

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

FCS_COP.1.1/SigGen The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

- RSA Digital Signature Algorithm,
- Elliptic Curve Digital Signature Algorithm

]

and cryptographic key sizes [

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

]

that meet the following: [

- For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,
- For ECDSA schemes implementing [P-256, P-384, P-521] curves that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST Recommended" curves; or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3, "IT Security

techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.

].

Application Note: This SFR has been modified by TD0921.

FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

FCS_COP.1.1/Hash The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [*SHA-1, SHA-256, SHA-384, SHA-512*] ~~and cryptographic key sizes [assignment: cryptographic key sizes]~~ and **message digest sizes [160, 256, 384, 512] bits** that meet the following: *ISO/IEC 10118-3:2004*.

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

FCS_COP.1.1/KeyedHash The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, implicit] 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"*.

FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [CTR_DRBG (AES)].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[*1*] software-based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1.1 The TSF shall implement the HTTPS protocol that complies with RFC 2818.

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS ~~protocol~~ using TLS.

FCS_SSH_EXT.1 SSH Protocol

FCS_SSH_EXT.1.1 The TOE shall implement *SSH* acting as a [server] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [5647, 5656, 6668, 8268, 8308, 8332] and [*no other standard*].

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [

- "password" (RFC 4252)
- "publickey" (RFC 4252): [
    - ssh-rsa (RFC 4253)

- rsa-sha2-256 (RFC 8332)
- rsa-sha2-512 (RFC 8332)
- ecdsa-sha2-nistp256 (RFC 5656)
- ecdsa-sha2-nistp384 (RFC 5656)
- ecdsa-sha2-nistp521 (RFC 5656)]

] and no other methods.

FCS_SSH_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*262144 bytes*] in an SSH transport connection are dropped.

FCS_SSH_EXT.1.4 The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms: [

- aes128-cbc (RFC 4253)
- aes256-cbc (RFC 4253)
- aes256-gcm@openssh.com (RFC 5647)

] and no other mechanisms.

FCS_SSH_EXT.1.5 The TSF shall protect data in transit from modification, deletion, and insertion using: [

- hmac-sha2-256 (RFC 6668)
- hmac-sha2-512 (RFC 6668)
- implicit

] and no other mechanisms.

FCS_SSH_EXT.1.6 The TSF shall establish a shared secret with its peer using: [

- diffie-hellman-group14-sha256 (RFC 8268),
- diffie-hellman-group16-sha512 (RFC 8268),
- diffie-hellman-group18-sha512 (RFC 8268),
- ecdh-sha2-nistp256 (RFC 5656),
- ecdh-sha2-nistp384 (RFC 5656),
- ecdh-sha2-nistp521 (RFC 5656),

] and no other mechanisms.

FCS_SSH_EXT.1.7 The TSF shall use *SSH KDF* as defined in [

- RFC 4253 (Section 7.2),

- RFC 5656 (Section 4)

] to derive the following cryptographic keys from a shared secret: *session keys*.

FCS_SSH_EXT.1.8 The TSF shall ensure that [

- a rekey of the session keys

] occurs when any of the following thresholds are met:

- one hour connection time
- no more than one gigabyte of transmitted data, or
- no more than one gigabyte of received data.

FCS_SSHS_EXT.1 SSH Protocol - Server

FCS_SSHS_EXT.1.1 The TSF shall authenticate itself to its peer (SSH Client) using: [

- rsa-sha2-256 (RFC 8332)
- ecdsa-sha2-nistp384 (RFC 5656)
- ecdsa-sha2-nistp521 (RFC 5656)

].

FCS_TLSC_EXT.1 TLS Client Protocol

FCS_TLSC_EXT.1.1 The TSF shall implement [TLS 1.3 (RFC 8446), TLS 1.2 (RFC 5246)] supporting the following ciphersuites: [

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 8422
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 8422
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384

] and no other ciphersuites.

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

FCS_TLSC_EXT.1.3 The TSF shall not establish a trusted channel if the server certificate is invalid [

- without any administrator override mechanism].

FCS_TLSC_EXT.1.4 The TSF shall [present the Supported Groups Extension with the following curves/groups: [secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192] and no other curves/groups] in the Client Hello.

FCS_TLSC_EXT.1.5 The TSF shall [

- present the signature_algorithms extension with support for the following algorithms: [
  - rsa_pkcs1 with sha256(0x0401),
  - rsa_pkcs1with sha384(0x0501),
  - rsa_pkcs1 with sha512(0x0601),
  - ecdsa_secp256r1 with sha256(0x0403),
  - ecdsa_secp384r1 with sha384(0x0503),
  - ecdsa_secp521r1 with sha512(0x0603),
  - rsa_pss_rsae with sha256(0x0804),
  - rsa_pss_rsae with sha384(0x0805),
  - rsa_pss_rsae with sha512(0x0806),
  - rsa_pss_pss with sha256(0x0809),
  - rsa_pss_pss with sha384(0x080a),
  - rsa_pss_pss with sha512(0x080b)
  -] and no other algorithms;

].

FCS_TLSC_EXT.1.6 The TSF [does not provide] the ability to configure the list of supported ciphersuites as defined in FCS_TLSC_EXT.1.1.

FCS_TLSC_EXT.1.7 The TSF shall prohibit the use of the following extensions:

- Early data extension
- Post-handshake client authentication according to RFC 8446, Section 4.2.6.

FCS_TLSC_EXT.1.8 The TSF shall [not use PSK's].

FCS_TLSC_EXT.1.9 The TSF shall [reject [TLS 1.2, TLS 1.3] renegotiation attempts].

FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

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

FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1 The TSF shall implement [TLS 1.3 (RFC 8446), TLS 1.2 (RFC 5246)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- 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_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_AES_128_GCM_SHA256
- TLS_AES_256_GCM_SHA384

] and no other ciphersuites.

FCS_TLSS_EXT.1.2 The TSF shall authenticate itself using X.509 certificate(s) using [RSA with key size [2048, 4096] bits; ECDSA over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves].

FCS_TLSS_EXT.1.3 The TSF shall perform key exchange using: [

- EC Diffie-Hellman key agreement over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves].

FCS_TLSS_EXT.1.4 The TSF shall support [no session resumption].

FCS_TLSS_EXT.1.5 The TSF [does not provide] the ability to configure the list of supported ciphersuites as defined in FCS_TLSS_EXT.1.1.

FCS_TLSS_EXT.1.6 The TSF shall prohibit the use of the following extensions:

- Early data extension

FCS_TLSS_EXT.1.7 The TSF shall [not use PSKs].

FCS_TLSS_EXT.1.8 The TSF shall [reject [TLS 1.2, TLS 1.3] renegotiation attempts].

FCS_IPSEC_EXT.1 IPsec Protocol

FCS_IPSEC_EXT.1.1 The TSF shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2 The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched and discards it.

FCS_IPSEC_EXT.1.3 The TSF shall implement [tunnel mode, transport mode].

FCS_IPSEC_EXT.1.4 The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms **[AES-CBC-256 (specified in RFC 3602), AES-GCM-256 (specified in RFC 4106)] and [no other algorithm]** together with a Secure Hash Algorithm (SHA)-based HMAC [HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512].

Application Note: *This element has been modified from its definition in the NDcPP by mandating either 128 or 256 bit key sizes for AES-CBC or AES-GCM, thereby disallowing for the sole selection of 192 bit key sizes. When an AES-CBC algorithm is selected, at least one SHA-based HMAC must also be chosen. If only an AES-GCM algorithm is selected, then a SHA-based HMAC is not*

required since AES-GCM satisfies both confidentiality and integrity functions. IPsec may use a truncated version of the SHA-based HMAC functions contained in the selections. Where a truncated output is used, this is described in the TSS.

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [

- IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109, [RFC 4304 for extended sequence numbers], and [RFC 4868 for hash functions]
- IKEv2 as defined in RFC 7296 [with mandatory support for NAT traversal as specified in RFC 7296, Section 2.23], and [RFC 4868 for hash functions]].

Application Note: This SFR element has been modified by TD0824.

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [IKEv1, IKEv2] protocol uses the cryptographic algorithms [AES-CBC-256 (specified in RFC 3602)].

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [

- IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on [
  - length of time, where the time values can be configured between [*120 seconds*] and [*24 hours*]];
- IKEv2 SA lifetimes can be configured by a Security Administrator based on [
  - length of time, where the time values can be configured between [*120 seconds*] and [*24 hours*]]].

Application Note: This SFR has been modified by TD0868.

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [

- IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on [
  - number of bytes
  - length of time, where the time values can be configured between [*120 seconds*] and [*8 hours*]];
- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [
  - number of bytes
  - length of time, where the time values can be configured between [*120 seconds*] and [*8 hours*]]].

Application Note: This SFR has been modified by TD0868.

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [*224, 256, 384*] bits.

FCS_IPSEC_EXT.1.10 The TSF shall generate nonces used in [IKEv1, IKEv2] exchanges of length [

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

FCS_IPSEC_EXT.1.11 The TSF shall ensure that IKE protocols implement DH Groups

- **19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and** [
- [14 (2048-bit MODP), 15 (3072-bit MODP), 16 (4096-bit MODP)] according to RFC 3526
- [**no other DH Groups**] according to RFC 5114

].

Application Note: *This element has been modified from its definition in the NDcPP by mandating DH groups 19 and 20, both of which are selectable in the original definition of the element. Any groups other than 19 and 20 may be selected by the ST author but they are not required for conformance to this PP-Module.*

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 [IKEv1 Phase 1, 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 [IKEv1 Phase 2, IKEv2 CHILD_SA] connection.

FCS_IPSEC_EXT.1.13 The TSF shall ensure that [**IKEv1, IKEv2]** protocols perform peer authentication using [RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [Pre-shared Keys that conform to RFC 8784].

Application Note: This SFR is modified by MOD_VPNGW_V1.3 from its definition in the Base-PP by adding new selections for IKE versions and authentication methods.

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), [no other reference identifier types].**

Application Note: *This PP-Module requires DN to be supported for certificate reference identifiers at minimum. Other selections may be made by the ST author but they are not required for conformance to this PP-Module.*

FCS_NTP_EXT.1 NTP Protocol

FCS_NTP_EXT.1.1 The TSF shall use only the following NTP version(s) [NTP v4 (RFC 5905)].

FCS_NTP_EXT.1.2 The TSF shall update its system time using [

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

].

FCS_NTP_EXT.1.3 The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

FCS_NTP_EXT.1.4 The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

5.3.3 User data protection (FDP)

FDP_RIP.2 Full Residual Information Protection

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

5.3.4 Identification and Authentication (FIA)

FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1-10*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password***.**

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

FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1.1 The TSF shall provide the following password management capabilities for administrative passwords:

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

b) Minimum password length shall be *configurable to between [8] and [64] characters.*

FIA_PSK_EXT.1 Pre-Shared Key Composition

FIA_PSK_EXT.1.1 The TSF shall be able to use pre-shared keys for IPsec and [IKEv2].

FIA_PSK_EXT.1.2 The TSF shall be able to accept the following as pre-shared keys: [generated bit-based] keys.

Application Note: This SFR is dependent on selections made in FCS_IPSEC_EXT.1.13.

FIA_PSK_EXT.2 Generated Pre-Shared Keys

FIA_PSK_EXT.2.1 The TSF shall be able to [

- accept externally generated pre-shared keys

]

Application Note: This SFR is dependent on selections made in FIA_PSK_EXT.1.

FIA_UAU.7 Protected Authentication Feedback (Refinement)

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**.

FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1.1 The TSF shall allow the following actions prior to requiring the non-TOE entity to initiate the identification and authentication process:

- Display the warning banner in accordance with FTA_TAB.1;
- [no other actions].

FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

FIA_UIA_EXT.1.3 The TSF shall provide the following remote authentication mechanisms [Web GUI password, SSH password, SSH public key] and [no other mechanism]. The TSF shall provide the following local authentication mechanisms [password-based].

FIA_UIA_EXT.1.4 The TSF shall authenticate any administrative user's claimed identity according to each authentication mechanism specified in FIA_UIA_EXT.1.3.

Application Note: This SFR has been modified by TD0900.

FIA_X509_EXT.1/Rev X.509 Certificate Validation

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

- RFC 5280 certificate validation and certificate path validation **supporting a minimum path length of three certificates**.
- The certificate path must terminate with a trusted CA certificate 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 [a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
    - *Server certificates presented for DTLS/TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
    - *Client certificates presented for DTLS/TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
    - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

FIA_X509_EXT.1.2/Rev The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

FIA_X509_EXT.2 X.509 Certificate Authentication

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for **IPsec and [HTTPS, TLS]**, and [[*support for client-side certificates for TLS mutual authentication with a FortiAnalyzer Audit Server*]].

Application Note: *The Base-PP allows the ST author to specify the TSF's use of X.509 certificates. Because this PP-Module mandates IPsec functionality, the SFR has been refined to force the inclusion of it. Other functions specified by the Base-PP may be chosen without restriction.*

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].

FIA_X509_EXT.3 X.509 Certificate Requests

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

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

5.3.5 Security management (FMT)

FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

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

FMT_MOF.1/Functions Management of Security Functions Behaviour

FMT_MOF.1.1/Functions The TSF shall restrict the ability to [modify the behaviour of] the functions [transmission of audit data to an external IT entity] to *Security Administrators*.

FMT_MOF.1/Services Management of Security Functions Behaviour

FMT_MOF.1.1/Services The TSF shall restrict the ability to **start and stop** ~~the functions~~ **services** to *Security Administrators*.

FMT_MTD.1/CoreData Management of TSF Data

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

FMT_MTD.1/CryptoKeys Management of TSF Data

FMT_MTD.1.1/CryptoKeys The TSF shall restrict the ability to [[*manage*]] the [*cryptographic keys **and certificates used for VPN operation***] to [*Security Administrators*].

Application Note: *This SFR, defined in the NDcPP as selection-based, is mandated for inclusion in this PP-Module because the refinements to FMT_SMF.1 mandate its inclusion. Note that it is also refined to refer specifically to keys and certificates used for VPN operation.*

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 The TSF shall be capable of performing the following management functions:

- *Ability to administer the TOE remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the remote session inactivity time before session termination;*
- *Ability to update the TOE, and to verify the updates using* digital signature *capability prior to installing those updates;*
- [
    - Ability to start and stop services;
    - Ability to configure local audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full; changes to local audit storage size);
    - Ability to modify the behaviour of the transmission of audit data to an external IT entity;
    - Ability to manage the cryptographic keys;
    - Ability to configure the cryptographic functionality;
    - Ability to configure the lifetime for IPsec SAs;
    - Ability to set the time which is used for time-stamps;
    - Ability to configure NTP;
    - Ability to configure the reference identifier for the peer;
    - Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
    - Ability to generate Certificate Signing Request (CSR) and process CA certificate response;
    - Ability to administer the TOE locally;
    - Ability to configure the local session inactivity time before session termination or locking;
    - Ability to configure the authentication failure parameters for FIA_AFL.1;
    - Ability to manage the trusted public keys database;].

Application Note: This SFR has been modified by TD0880.

FMT_SMF.1/FFW Specification of Management Functions

FMT_SMF.1.1/FFW The TSF shall be capable of performing the following management functions:

- *Ability to configure firewall rules;*

FMT_SMF.1/VPN Specification of Management Functions

FMT_SMF.1.1/VPN The TSF shall be capable of performing the following management functions [

- *Definition of packet filtering rules*
- *Association of packet filtering rules to network interfaces*

- *Ordering of packet filtering rules by priority*

[

- *No other capabilities]]*.

FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- *Security Administrator*.

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE remotely*

are satisfied.

5.3.6 Protection of the TSF (FPT)

FPT_APW_EXT.1 Protection of Administrator Passwords

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

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

FPT_TST_EXT.1 TSF Testing

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

- During initial start-up (on power on) to verify the integrity of the TOE firmware and software;
- Prior to providing any cryptographic service and [on-demand] to verify correct operation of cryptographic implementation necessary to fulfil the TSF;
- [start-up] self-tests *[CPU and Memory BIOS self-tests, Boot loader image verification, FIPS 140-2 Cryptographic Known Answer Tests (KAT)]* to demonstrate the correct operation of the TSF: **noise source health tests.**

Application Note: *This SFR element has been modified by TD0824 and TD0836.*

FPT_TST_EXT.1.2 The TSF shall respond to [all failures] by [entering a maintenance mode].

Application Note: *This SFR element has been modified by TD0824.*

FPT_TST_EXT.3 Self-Test with Defined Methods

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*].

FPT_TST_EXT.3.2 The TSF shall execute the self-testing through [*a TSF-provided cryptographic service specified in FCS_COP.1/**SigGen***].

FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software and [no other TOE firmware/software version].

FPT_TUD_EXT.1.2 The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and [no other update mechanism].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a digital signature mechanism and [no other mechanisms] prior to installing those updates.

Application Note: *This SFR element has been modified by TD0824.*

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

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

FPT_STM_EXT.1 Reliable Time Stamps

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

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

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

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

Application Note: *This SFR defines the expected TSF response to failures of the self-tests defined in the Base-PP.*

5.3.7 TOE access (FTA)

FTA_SSL_EXT.1 TSF-initiated Session Locking

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

- terminate the session

] after a Security Administrator-specified time period of inactivity.

FTA_SSL.3 TSF-initiated Termination

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

FTA_SSL.4 User-initiated Termination

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

FTA_TAB.1 Default TOE Access Banners

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

5.3.8 Trusted path (FTP)

FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1 The TSF shall **be capable of using [TLS] to** provide a **trusted** communication channel between itself and ~~another trusted IT product~~ **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 ~~modification or~~ disclosure **and detection of modification of the channel data**.

FTP_ITC.1.2 The TSF shall permit [the TSF] to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for ***[audit server***].

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

FTP_ITC.1.1/VPN The TSF shall **be capable of using IPsec to** provide a communication channel between itself and **authorized IT entities supporting VPN communications** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from **disclosure and detection of modification of the channel data**.

FTP_ITC.1.2/VPN The TSF shall permit [*the authorized IT entities*] to initiate communication via the trusted channel.

FTP_ITC.1.3/VPN The TSF shall initiate communication via the trusted channel for [remote VPN gateways or peers].

FTP_TRP.1/Admin Trusted Path

FTP_TRP.1.1/Admin The TSF shall **be capable of using** [**SSH, HTTPS**] **to** provide a communication path between itself and **authorized** remote **Administrators** ~~users~~ that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from disclosure **and provides detection of modification of the channel data.**

FTP_TRP.1.2/Admin The TSF shall permit remote **Administrators** ~~users~~ to initiate communication via the trusted path.

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

5.3.9 Stateful traffic filtering (FFW)

FFW_RUL_EXT.1 Stateful Traffic Filtering

FFW_RUL_EXT.1.1 The TSF shall perform stateful traffic filtering on network packets processed by the TOE.

FFW_RUL_EXT.1.2 The TSF shall allow the definition of stateful traffic filtering rules using the following network protocol fields:

- *ICMPv4*
  - *Type*
  - *Code*
- *ICMPv6*
  - *Type*
  - *Code*
- *IPv4*
  - *Source address*
  - *Destination Address*
  - *Transport Layer Protocol*
- *IPv6*
  - *Source address*
  - *Destination Address*
  - *Transport Layer Protocol*
  - *[no other field]*
- *TCP*
  - *Source Port*
  - *Destination Port*
- *UDP*
  - *Source Port*
  - *Destination Port*

and distinct interface.

FFW_RUL_EXT.1.3 The TSF shall allow the following operations to be associated with stateful traffic filtering rules: permit or drop with the capability to log the operation.

FFW_RUL_EXT.1.4 The TSF shall allow the stateful traffic filtering rules to be assigned to each distinct network interface.

FFW_RUL_EXT.1.5 The TSF shall:

a) accept a network packet without further processing of stateful traffic filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [ICMP] based on the following network packet attributes:

1. *TCP: source and destination addresses, source and destination ports, sequence number, Flags;*
2. *UDP: source and destination addresses, source and destination ports;*
3. [*ICMP: source and destination addresses, type,* [*code*]].

b) Remove existing traffic flows from the set of established traffic flows based on the following: [session inactivity timeout, completion of the expected information flow].

FFW_RUL_EXT.1.6 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].

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

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

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

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

FFW_RUL_EXT.1.8 The TSF shall process the applicable stateful traffic filtering rules in an administratively defined order.

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

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

5.3.10 Packet filtering (FPF)

FPF_RUL_EXT.1 Packet Filtering Rules

FPF_RUL_EXT.1.1 The TSF shall perform packet filtering on network packets processed by the TOE.

FPF_RUL_EXT.1.2 The TSF shall allow the definition of packet filtering rules using the following network protocols and protocol fields:

- *IPv4 (RFC 791)*
  - *Source address*
  - *Destination address*
  - *Protocol*
- *IPv6 (RFC 8200)*
  - *Source address*
  - *Destination address*
  - *Next header (protocol)*
- *TCP (RFC 793)*
  - *Source port*
  - *Destination port*
- *UDP (RFC 768)*
  - *Source port*
  - *Destination port*

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.

FPF_RUL_EXT.1.4 The TSF shall allow the packet filtering rules to be assigned to each distinct network interface.

FPF_RUL_EXT.1.5 The TSF shall process the applicable packet filtering rules (as determined in accordance with FPF_RUL_EXT.1.4) in the following order: [*Administrator-defined*].

FPF_RUL_EXT.1.6 The TSF shall drop traffic if a matching rule is not identified.

5.4 Assurance Requirements

19 The TOE security assurance requirements are summarized in Table 18.

Table 18: Security Assurance Requirements

Assurance Class	Assurance Components
Security Target (ASE)	Conformance Claims (ASE_CCL.1)
	Extended Components Definition (ASE_ECD.1)
	ST Introduction (ASE_INT.1)
	Security Objectives for the Operational Environment (ASE_OBJ.1)
	Stated Security Requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE Summary Specification (ASE_TSS.1)
Development (ADV)	Basic Functional Specification (ADV_FSP.1)
Guidance Documents (AGD)	Operational User Guidance (AGD_OPE.1)
	Preparative Procedures (AGD_PRE.1)
Life Cycle Support (ALC)	Labelling of the TOE (ALC_CMC.1)
	TOE CM Coverage (ALC_CMS.1)
Tests (ATE)	Independent Testing – Conformance (ATE_IND.1)
Vulnerability Assessment (AVA)	Vulnerability Survey (AVA_VAN.1)

20 In accordance with CPP_ND, the following refinement is made to ASE:

a) **ASE_TSS.1.1C Refinement:** The TOE summary specification shall describe how the TOE meets each SFR. **In the case of entropy analysis, the TSS is used in conjunction with required supplementary information on Entropy.**

6 TOE Summary Specification

6.1 Security Audit

SFRs:	FAU_GEN.1, FAU_GEN.1/VPN, FAU_GEN.2, FAU_STG_EXT.1

21 The TOE generates audit records as identified in section 5.3.1.

22 For each auditable event, the TOE records the date and time of the event, subject identity (i.e. administrative user), type of event and/or reaction and (where applicable) the success or failure of the event.

23 Specific audit events required by FAU_GEN.1 and generated by the TOE can be found in Table 16. Specific audit events required by FAU_GEN.1/VPN and generated by the TOE can be found in Table 17.

24 Local log files can only be deleted via the CLI by an authorized administrator. No editing of log data is permitted.

25 In the evaluated configuration, the TOE is configured to transmit log data to an external FortiAnalyzer platform. All log data is buffered in system memory prior to immediately being transmitted to FortiAnalyzer in the event communication is interrupted between the TOE and the remote audit server. This process occurs in real-time and is transmitted via TLS.

26 The TOE provides a single standalone component that stores audit data locally in a non-persistent manner which consists of a hardcoded minimum file size of 10mb of the TOE model's system memory up to a configurable maximum of 5% of total system RAM (up to a value of 4GB). The default maximum is 1% of system RAM (see Table 4).

27 If the local storage for audit logs is filled, the oldest stored logs will be deleted in a First-In-First-Out (FIFO) order to allow for the saving of new event.

28 The TOE provides administrators with a mechanism to view audit records locally.

29 If one of the TOE interfaces is overwhelmed by traffic, then blocking the anomaly occurs as specified in the TOE DoS Policy. Packets are dropped when an Administrator-configurable threshold is met.

30 The following information is logged as a result of the Security Administrator generating/importing or deleting cryptographic keys:

a) **Generate SSH key-pair.** Action and key reference (Name or ID).

b) **Generate CSR.** Action and key reference (Name or ID).

c) **Import Certificate.** Action and key reference (Name or ID).

d) **Import CA Certificate.** Action and key reference (Name or ID).

6.2 Cryptographic Support

SFRs:	FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1/KeyedHash, FCS_RBG_EXT.1, FCS_CKM.1, FCS_CKM.1/IKE, FCS_CKM.2, FCS_CKM.4

31 The following tables identify the cryptographic algorithms and methods implemented by the TOE. CAVP certificates are identified at Annex B: CAVP Certificates.

Table 19: Key Generation Methods

Method	Key Size (bits)	Curves	Standard
RSA	2048, 4096	N/A	FIPS 186-5, Appendix A.1 The TOE implements all “shall” and “should” statements and does not implement any “"shall not" " or "should not" statements. Details of “should” statements: • Pg. 64 & 65 – If an error is encountered during the generation process invalid values are returned.
Elliptic-curve	256 384 521	P-256 P-384 P-521	FIPS 186-5, Appendix A.2 The TOE implements all “shall” and “should” statements and does not implement any “"shall not" " or "should not" statements. Details of “should” statements: • Pg. 63 – If an error is encountered during the generation process invalid values are returned.
FFC Schemes using safe-prime groups	2048, 3072, 4096, 8192 ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192	N/A	RFC 3526 - Section 3, Section 4, Section 5. RFC 7919 – Appendix A.

32 The TOE generates asymmetric cryptographic keys used for IKE peer authentication in accordance with the algorithms claimed in FCS_CKM.1.1/IKE and specified cryptographic key sizes equivalent to or greater than a symmetric key strength of 256 bits.

Table 20: Key Establishment Methods

Method	Usage	Services
Elliptic-curve schemes	Used in TLS, IPsec, and SSH. TOE is both sender and receiver.	TLS Client (Audit Server) TLS/HTTPS Server (GUI) IPsec (VPN) SSH Server (Admin CLI)
FFC Schemes using safe-prime groups	Used in IPsec. The TOE meets RFC 3526 Section 3, Section 4, and Section 5 by implementing DH Groups 14 (2048-bit MODP), 15 (3072-bit MODP), and 16 (4096-bit MODP). Used in TLS Client. The TOE implements the ffdhe safe prime groups listed in RFC 7919 Appendix A.	IPsec (VPN) TLS Client (Audit Server) SSH Server (Admin CLI)

Method	Usage	Services
	Used in SSH. The TOE implements DH Groups 14 (2048-bit MODP), 16 (4096-bit MODP), and 18 (8192-bit MODP) per RFC 3526.	

Table 21: Cryptographic Methods

Operation	Algorithm	Key size(bits)	Digest size	Block size	Standard(s)
Encryption and decryption	AES in CBC or GCM modes	128 256	n/a	128	ISO 18033-3 ISO 10116 ISO 19772
Signature generation and verification	RSA	2048 3072 4096	n/a	n/a	FIPS 186-5 ISO/IEC 9796-2
	ECDSA	256	n/a	n/a	FIPS 186-4
		384 521	n/a	n/a	FIPS 186-5 ISO/IEC 14888-3
Hashing	SHA	n/a	160 256 384 512	512 512 1024 1024	ISO/IEC 10118-3:2004
Keyed-hash message authentication	HMAC-SHA	160 256 384 512	160 256 384 512	512 512 1024 1024	ISO/IEC 9797-2:2011 Section 7
Random bit generation	CTR_DRBG	n/a	n/a	n/a	ISO/IEC 18031:2011

6.2.1 Hash Usage

33 SHA is implemented in the following functions of the TOE:

a) TLS;

b) SSH;

c) IPsec;

d) Digital signature verification as part of trusted update validation;

6.2.2 Keys and CSPs

34 The TOE only stores keys in memory, either in RAM or Flash memory. The TOE provides the following zeroization methods for cryptographic keys and other material:

a) **Volatile memory (SDRAM).** The TOE performs a single direct overwrite consisting of zeroes, followed by a read-verify. If the read-verification of the overwritten data fails, the process repeats.

b) **Non-volatile flash memory (Flash RAM).** The TOE performs a single, direct overwrite consisting of zeroes, which is followed by a followed by a read-verify. If the read-verification fails, the process repeats.

35 Zeroization of cryptographic keys is performed via the OS kernel and invoked via the Command Line Interface (CLI). There are no interfaces available to users, or otherwise enabled roles, that are designed specifically for the purpose of viewing keys and passwords in plaintext.

36 The following table lists the keys/CSPs used by the TOE, their storage location and format and their associated zeroization method, per the description above.

Table 22: Keys and CSPs

Key/CSP	Storage location and method	Usage	Zeroization
IPSec Session Authentication Key	Plaintext in RAM	IPsec peer-to-peer authentication using HMAC-SHA-256	Overwritten with zeroes when no longer needed.
IPSec Session Encryption Key	Plaintext in RAM	VPN traffic encryption/decryption using AES (128-,256-bit)	Overwritten with zeroes when no longer needed.
IKE SKEYSEED	Plaintext in RAM	Used to generate IKE protocol keys	Overwritten with zeroes when no longer needed.
IKE Authentication Key	Plaintext in RAM	IKE peer-to-peer authentication using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512	Overwritten with zeroes when no longer needed.
IKE Key Generation Key	Plaintext in RAM	IPsec SA keying material	Overwritten with zeroes when no longer needed.
IKE Session Encryption Key	Plaintext in RAM	Encryption of IKE peer-to-peer key negotiation using AES (128, 256-bit)	Overwritten with zeroes when no longer needed.
IKE Pre-Shared Key	Encrypted (AES-128) in Flash	Used to generate IKE protocol keys	Overwritten with zeroes when no longer needed.
RSA Private Key	Plaintext in Flash (generated with CSR or imported)	RSA private key used in IKE (2048-bit and 4096-bit signatures)	Overwritten with zeroes when no longer needed.

Key/CSP	Storage location and method	Usage	Zeroization
ECDSA Private Key	Plaintext in Flash (generated with CSR or imported)	ECDSA private key used in IKE (signatures using P-256, -384 and -521 curves)	Overwritten with zeroes when no longer needed.
Diffie-Hellman Keys	Plaintext in RAM	Key agreement and key establishment	Overwritten with zeroes when no longer needed.
EC Diffie-Hellman Keys	Plaintext in RAM	Key agreement and key establishment	Overwritten with zeroes when no longer needed.
Firmware Update Key	Plaintext in RAM	Verification of firmware integrity when updating to new firmware versions using RSA public key	Overwritten with zeroes when no longer needed.
HTTPS/TLS Server/Host Key	Plaintext in Flash	RSA & ECDSA private keys used in the HTTPS/TLS protocols	Overwritten with zeroes when no longer needed.
HTTPS/TLS Client/Host Key	Plaintext in Flash	RSA & ECDSA private keys used in the HTTPS/TLS protocols	Overwritten with zeroes when no longer needed.
HTTPS/TLS Session Authentication Key	Plaintext in RAM	HMAC SHA-1, -256 or -384 key used for HTTPS/TLS session authentication	Overwritten with zeroes when no longer needed.
HTTPS/TLS Session Encryption Key	Plaintext in RAM	AES (128-, 256-bit) key used for HTTPS/TLS session encryption	Overwritten with zeroes when no longer needed.
SSH Server/Host Key	Plaintext in Flash	RSA private key used in the SSH protocol (key establishment, 2048 -bit)	Overwritten with zeroes when no longer needed.
SSH Session Authentication Key	Plaintext in RAM	HMAC-SHA2-256, or HMAC-SHA2-512 key used for SSH session authentication	Overwritten with zeroes when no longer needed.
SSH Session Encryption Key	Plaintext in RAM	AES (256-bit) key used for SSH session encryption	Overwritten with zeroes when no longer needed.
SSH Public Key	Plaintext in Flash	RSA public key used in SSH protocol (key establishment, 2048 -bit)	Overwritten with zeroes when no longer needed.
Locally Stored Passwords	AES-128 encrypted in configuration file (in FIPS mode)	User authentication	Overwritten with zeroes when no longer needed.

Key/CSP	Storage location and method	Usage	Zeroization
Configuration Encryption Key	Plaintext in Flash	AES 128-bit key used to encrypt CSPs on the Boot device	Overwritten with zeroes when no longer needed.

6.2.3 Entropy and DRBG

37 As shown in Table 4 (Entropy column), The TOE makes use of JitterEnt as the software-based entropy source for all claimed models.

38 In all models, the TOE contains a NIST 800-90A approved Kernel CTR_DRBG that is seeded from the software-based entropy source. Entropy from the noise source is extracted, conditioned, and used to seed the DRBG with a calculated 256 bits of full entropy.

39 Additional detail regarding the DRBG implementation and use of JitterEnt is provided with the proprietary Entropy Description.

6.3 HTTPS/TLS

SFRs:	FCS_HTTPS_EXT.1, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.1

6.3.1 HTTPS

40 The TOE web GUI is accessed via an HTTPS connection using the TLS implementation described by FCS_TLSS_EXT.1. The TOE does not use HTTPS in a client capacity. The TOE's HTTPS protocol complies with RFC 2818.

41 RFC 2818 specifies HTTP over TLS. The majority of RFC 2818 is spent on discussing practices for validating endpoint identities and how connections must be set up and torn down. The TOE web GUI operates on an explicit port designed to natively speak TLS: it does not attempt STARTTLS or similar multi-protocol negotiation which is described in section 2.3 of RFC 2818. The web server uses a variant of OpenSSL which attempts to send closure Alerts prior to closing a connection in accordance with section 2.2.2 of RFC 2818.

6.3.2 TLS Server

42 The TOE operates as a TLS server for the web GUI trusted path.

43 The server only allows TLS protocol version 1.2 per RFC 5246 and 1.3 per RFC 8446 (rejecting any other protocol version, including SSL 2.0, SSL 3.0, TLS 1.0, TLS 1.1 and any other unknown TLS version string supplied).

44 TLS version 1.2 is restricted to the following ciphersuites:

a) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

b) TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

c) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

45 TLS version 1.3 is restricted to the following ciphersuites:

a) TLS_AES_128_GCM_SHA256

b) TLS_AES_256_GCM_SHA384

46 Ciphersuites are not configurable by users or administrators.

47 The TOE is restricted to authenticating itself to TLS clients using X.509 certificates with RSA key sizes of 2048-bits and 4096-bits, and ECDSA over NIST curves secp256r1, secp384r1, and secp521r1.

48 The TLS server is capable of negotiating ciphersuites that include ECDHE key agreement schemes. The ECDHE key agreement parameters use secp256r1, secp384r1, and secp521r1 and are hardcoded into the server.

49 The TOE does not support session resumption and rejects all TLS renegotiation attempts.

50 The TOE does not make use of pre-shared keys for TLS.

6.3.3 TLS Client

51 The TOE operates as a TLS client for the trusted channel with the FortiAnalyzer Server.

52 The TOE restricts the use of TLS to versions 1.2 and 1.3 only.

53 TLS version 1.2 ciphersuites are restricted to the following:

a) TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

b) TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246

c) TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268

d) TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246

e) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 8422

f) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

g) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 8422

h) TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

i) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422

j) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289

k) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422

l) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289

54 TLS version 1.3 ciphersuites are restricted to the following:

a) TLS_AES_128_GCM_SHA256

b) TLS_AES_256_GCM_SHA384

55 Ciphersuites are not configurable by users or administrators.

56 The reference identifier for the FortiAnalyzer Server is configured by the administrator using the web GUI (IP address) or CLI (IP address or DNS name).

57 When the TLS client receives an X.509 certificate from the server, the client will compare the reference identifier with the established Subject Alternative Names (SANs) in the certificate. If a SAN is available and does not match the reference identifier, then the verification fails and the channel is terminated. If there are no SANs of the correct type (IP address or DNS name) in the certificate, then the TOE will compare the reference identifier to the CN (DNS name) in the certificate Subject. If there is no CN, then the verification fails and the channel is terminated. If the CN exists and does not match, then the verification fails and the channel is terminated. Otherwise, the reference identifier verification passes and additional verification actions can proceed. The TOE supports wildcards for DNS names in the CN and SAN.

58 The TLS client does not support certificate pinning or administrator override of certificate validation failures.

59 The TLS client will transmit the Supported Groups Extension in the Client Hello message by default with support for the following NIST curves/groups: secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, and ffdhe8192. The non-TOE server may choose to negotiate any of the supported elliptic curves or DHE ciphersuites from this set. If an unsupported DHE parameter set is returned in the Server Key Exchange, the TOE will terminate the connection. The TOE also presents the signature_algorithms extension by default with support for the following algorithms:

a) rsa_pkcs1 with sha256(0x0401)

b) rsa_pkcs1with sha384(0x0501)

c) rsa_pkcs1 with sha512(0x0601)

d) ecdsa_secp256r1 with sha256(0x0403)

e) ecdsa_secp384r1 with sha384(0x0503)

f) ecdsa_secp521r1 with sha512(0x0603)

g) rsa_pss_rsae with sha256(0x0804)

h) rsa_pss_rsae with sha384(0x0805)

i) rsa_pss_rsae with sha512(0x0806)

j) rsa_pss_pss with sha256(0x0809)

k) rsa_pss_pss with sha384(0x080a)

l) rsa_pss_pss with sha512(0x080b)

60 The TOE prohibits the use of the following extensions by default:

a) Early data extension

b) Post-handshake client authentication according to RFC 8446, Section 4.2.6.

61 The TOE does not make use of pre-shared keys for TLS.

62 The TOE rejects all renegotiation attempts for TLS v1.2 and v1.3.

63 The TOE supports presentation of an X.509v3 client certificate for authentication as required by the FortiAnalyzer Audit Server.

6.4 SSH

SFRs:	FCS_SSH_EXT.1, FCS_SSHS_EXT.1

64 The TOE implements SSH (as a server) in compliance with RFCs 4251, 4252, 4253, 4254, 5647, 5656, 6668, 8268, 8308, and 8332.

65 The TOE implements both password-based and public key client authentication and supports the following algorithms:

a) ssh-rsa per RFC 4253

b) rsa-sha2-256 per RFC 8332

c) rsa-sha2-512 per RFC 8332

d) ecdsa-sha2-nistp256 per RFC 5656

e) ecdsa-sha2-nistp384 per RFC 5656

f) ecdsa-sha2-nistp521 per RFC 5656

66 The TOE examines the size of each received SSH packet. If the packet is greater than 262144 bytes, it is automatically dropped.

67 The TOE utilizes aes128-cbc per RFC 4253, aes256-cbc per RFC 4253, and AES256-GCM@openssh.com per RFC 5647 for data in transit encryption via SSH, and rejects all other algorithms.

68 The TOE provides data integrity for data in transit through SSH connections via hmac-sha2-256 per RFC 6668, hmac-sha2-512 per RFC 6668, and implicit mechanisms and rejects all other algorithms.

69 The TOE supports the following algorithms for SSH key exchanges:

a) diffie-hellman-group14-sha256 (RFC 8268),

b) diffie-hellman-group16-sha512 (RFC 8268),

c) diffie-hellman-group18-sha512 (RFC 8268),

d) ecdh-sha2-nistp256 (RFC 5656),

e) ecdh-sha2-nistp384 (RFC 5656),

f) ecdh-sha2-nistp521 (RFC 5656),

70 The TOE implements SSH KDF per RFC4253 Section 7.2 and RFC5656 Section 4.

71 The TOE will re-key SSH session keys after 1 hour connection time or after no more than 1 gigabyte of data has been transmitted or received (whichever occurs first).

72 The TOE establishes a user identity by either verifying that the SSH client's present public key matches the one that is stored within the SSH server's authorized keys file, or by confirming the validity of the username and password presented.

6.5 IPsec

SFRs:	FCS_IPSEC_EXT.1

73 The TOE implements IPsec in accordance with RFC 4301.

74 Incoming packets are inspected against the session database. Sessions that match all the security attributes and do not exceed the TTL are automatically passed on to their destination and are sent via a VPN interface where applicable. Packets that do not match the attributes in the session database are then compared to the defined firewall rules for that interface identifier based on their unique numerical order. Packets that are permitted are passed to their destination, packets marked for logging are written to the audit log and packets marked for dropping are discarded.

75 The TOE permits three actions to be assigned to packet rules – BYPASS (allow the packet to flow through the TOE with no protection), DISCARD (drop the packet with no further processing) and PROTECT (encrypt the packet).

76 SPD entries are enforced in an administrator-defined order. If no rules matching the inbound traffic are present within the SPD, the default “no-match” rule will be applied.

77 The TOE can be configured to establish VPN connections in transport mode or tunnel mode.

78 The TOE implements the ESP protocol as defined in RFC 4303. The TOE implements AES-CBC-256 (per RFC 3602) and AES-GCM-256 (per RFC 4106) in conjunction with the following truncated versions of Secure Hash Algorithm-based HMAC algorithms to provide encryption services for ESP: HMAC-SHA-256-128, HMAC-SHA-384-192, and HMAC-SHA-512-256.

79 The TOE implements IKEv1 (as defined in RFCs 2407, 2408, 2409 and 4109) with RFC 4304 for extended sequence numbers and RFC 4868 for hash functions, and IKEv2 (as defined in RFC 7296), with mandatory support for NAT traversal as specified in RFC 7296 Section 2.23 and RFC 4868 for hash functions. IKE Peer-to-peer authentication uses HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512.

80 The TOE does not use aggressive mode for IKEv1 Phase 1 exchanges and only main mode is permitted in the evaluated configuration.

81 The TOE implements AES-CBC-256 (per RFC 3602) to provide payload encryption for IKEv1 and IKEv2.

82 The TOE permits the configuration of IKEv1 Phase 1 SA and IKEv2 SA lifetimes in seconds, between 120 and 86,400 (24 hours).

83 The TOE permits the configuration of IKEv1 Phase 2 SA and IKEv2 Child SA lifetimes in number of bytes between 5KB and 4GB, or seconds, between 120 and 28,800 (8 hours).

84 The TOE utilizes CTR-DRBG with AES (as specified in FCS_RBG_EXT.1) to generate the exponents used in IKE key exchanges, having the possible lengths of 224, 256 or 384 bits, corresponding to each of the supported DH groups. Nonces used in IKE are generated in this same way for negotiated PRF hashes. Nonce sizes are:

a) 128 bits for SHA-256;

b) 256 bits for SHA-384 and SHA-512.

85 The TOE supports Diffie-Hellman groups 14, 15, 16, 19, and 20. The specific group to be used for any given IPsec connection is specified in the IPsec policy configuration.

86 The TOE provides encryption algorithms with a strength of 256 bits for use in IKE and ESP exchanges. When negotiating Phase 2 (IKEv1) or CHILD_SA (IKEv2) ciphersuites, the TOE checks to ensure that the encryption strengths (in bits) for the selected algorithms are less than or equal to the encryption strengths of the algorithms selected for the Phase 1 (IKEv1) or IKE_SA (IKEv2) connection.

87 The TOE permits peer authentication via RSA or ECDSA public keys (X509v3 certificates that conform to RFC 4945) for IKEv1 and IKEv2, and pre-shared keys that conform to RFC 8784 for IKEv2.

88 When using certificates for peer authentication, the TOE will only establish a trusted channel to peers that provide a valid certificate. The TOE will compare the reference identifier of the peer against the reference identifier stored in the associated certificate. If the two values are not a match, the TOE will not establish the connection. The TOE supports Distinguished Name (DN) reference identifiers.

6.6 NTP

SFRs:	FCS_NTP_EXT.1

89 The TOE implements Network Time Protocol version 4 per RFC 5905 and uses SHA1 as the message digest algorithm to verify the authenticity of timestamp updates.

90 The TOE does not accept NTP timestamp updates from broadcast or multicast addresses.

91 The TOE supports a configuration of up to three NTP time sources in the operational environment.

6.7 Residual Data Protection

SFRs:	FDP_RIP.2

92 The TOE ensures that no information from previously processed information flows is transferred to subsequent information flows. This applies both to information that is input to the TOE from an external source and to information (e.g., padding bits) that might be added by the TOE during processing of the information from the external source. The removal of any previous residual

information is done through the zeroization of data when the memory structure is initially created and strict bounds checking on the data prior to it being assigned in memory.

6.8 Identification and Authentication

SFRs:	FIA_AFL.1, FIA_PMG_EXT.1, FIA_PSK_EXT.1, FIA_PSK_EXT.2, FIA_UAU.7, FIA_UIA_EXT.1

93 The TOE permits administrators to set a positive integer for failed remote authentication attempts. When this limit is met, the remote user must wait for a defined period of time before further authentication attempts can be made. The local console does not implement the lockout mechanism.

94 The TOE enforces a password policy. Administrative passwords are configurable to enforce a minimum password length of at least 8 characters and at most 64 characters and may be comprised of any combination of upper and lower case letters, numbers, and the following special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”.

95 Administrators connecting via a local connection (console) or remote (HTTPS or SSH) must provide a valid username and password or recognized pubic key to complete the authentication process. The TOE provides no feedback while authentication is in progress at the console. The logon process is as follows:

a) The local administrator connects to the TOE via the console port.

b) For remote connections, the remote administrator connects via SSH or the web GUI (HTTPS). Key exchange and session establishment actions take place;

c) The administrator is prompted for their username and password, which they enter (this step may be skipped if the TOE is configured to use public-key based authentication for SSH).

d) If the username and password provided is incorrect (or public key authentication fails), the administrator is presented with an error. See above for the TOE’s behavior if the number of unsuccessful attempts exceeds the defined threshold; or

e) If the username and password provided are correct (and/or public key authentication succeeds), the TOE shall end the logon process and give the administrator access to TOE functionality (a successful logon).

96 The TOE provides support for the use of externally generated 256 bit-based pre-shared keys that conform to RFC 8784 for IKEv2 only.

6.9 X509 Certificates

SFRs:	FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3

97 The TOE performs X.509 certificate validation at the following points:

a) TLS client validation of server certificates;

b) IPsec peer authentication;

c) When certificates are loaded into the TOE, such as when importing CAs, certificate responses and other device-level certificates (such as the web server certificate presented by the TOE TLS web GUI).

98 In all scenarios, certificates are checked for several validation characteristics:

a) If the certificate ‘notAfter’ date is in the past, then this is an expired certificate which is considered invalid;

b) The certificate chain must terminate with a trusted CA certificate;

c) Server certificates consumed by the TOE TLS client must have a ‘serverAuthentication’ extendedKeyUsage purpose;

99 A trusted CA certificate is defined as any certificate loaded into the TOE trust store that has, at a minimum, a basicConstraints extension with the CA flag set to TRUE.

100 Certificate revocation checking for the above scenarios is performed using a CRL as specified in RFC 5280 Section 6.3 and RFC 8603 Section 7.

101 As X.509 certificates are not used for trusted updates, firmware integrity self-tests, client authentication, or OCSP, the code-signing, clientAuthentication, and OCSP signing purpose is not checked in the extendedKeyUsage for related certificates.

102 The TOE has a trust store where root CA and intermediate CA certificates can be stored. The trust store is not cached: if a certificate is deleted, it is immediately untrusted. If a certificate is added to the trust store, it is immediately trusted for its given scope.

103 Revocation checking is performed on both leaf and intermediate CA certificates when a leaf certificate is presented to the TOE as part of the certificate chain during authentication.

104 The X.509 certificates for each of the given scenarios are validated using the certificate path validation algorithm defined in RFC 5280, which can be summarized as follows:

a) The public key algorithm and parameters are checked

b) The current date/time is checked against the validity period revocation status is checked

c) Issuer name of X matches the subject name of X+1

d) Name constraints are checked

e) Policy OIDs are checked

f) Policy constraints are checked; issuers are ensured to have CA signing bits

g) Path length is checked

h) Critical extensions are processed

105 If, during the trust chain verification activity, any certificate under review fails a verification check, then the certificate is deemed untrusted and the connection is rejected.

106 The TOE uses the leaf certificate presented by an external IT entity to authenticate the external IT entity. The TOE uses any presented and stored intermediate CA certificates to build a trust chain as described above. The certificates used by the TOE for this purpose are determined by those configured by the administrator. The client certificates used by the TOE for mutual authentication are also determined by those configured by the administrator.

107 As part of the verification process, CRL is used to determine whether the certificate is revoked or not. If the CRL cannot be obtained or otherwise accessed to verify the status of the certificate, the certificate is not accepted. CRLs are obtained from a web server over HTTP and are refreshed according to the following schedule:

a) By default they are refreshed based on the “next update” field in the CRL;

b) If the CRL update-interval in the TOE CLI is set to non-zero value (N), then it will refresh every N seconds.

108 Instructions for configuring the trusted IT entities to supply appropriate X.509 certificates are captured in the guidance documents.

109 For Certificate Signing Requests, the TOE conforms to RFC 2986 and provides the following information when generating the request: public key, Common Name, Organization, Organizational Unit, Country.

110 The TSF validates the chain of certificates from the Root CA upon receiving the CA certificate response.

6.10 Security Management

SFRs:	FMT_MOF.1/ManualUpdate, FMT_MOF.1/Functions, FMT_MOF.1/Services, FMT_MTD.1/CoreData, FMT_MTD.1/CryptoKeys, FMT_SMF.1, FMT_SMF.1/FFW, FMT_SMF.1/VPN, FMT_SMR.2

111 The TOE restricts the management functions in this section to the Security Administrator.

112 The TOE does not permit access to any functions (other than the warning/consent banner and authentication interface) prior to login.

113 The TOE permits the Security Administrator to manage (generate, import, delete) the following keys: IKE RSA and IKE ECDSA keys used for VPN operations, in addition to HTTPS/TLS and SSH server host key pairs, and TLS client key pairs.

114 The TOE defines a single role, which is that of the Security Administrator. The Security Administrator is able to start and stop the trusted path / trusted channels via the GUI (HTTPS), the local CLI, and remote CLI (SSH). The Security Administrator is able to perform the following functions via GUI interfaces, local CLI, and remote CLI:

- Ability to administer the TOE remotely;
- Ability to configure the access banner;
- Ability to configure the remote session inactivity time before session termination;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- Ability to start and stop services (and restart SSHD and HTTPSD services);
- Ability to configure local audit behaviour (changes to local audit storage size);
- Ability to modify the behaviour of transmission of audit records to an external IT entity (audit server);
- Ability to manage the cryptographic keys: Generate and delete cryptographic keys. In particular, a security administrator can generate and delete the cryptographic keys associated with CSRs;
- Ability to configure the cryptographic functionality;
- Ability to configure the lifetime for IPsec SA’s;
- Ability to set the time which is used for time-stamps, or configure NTP;
- Configure the IPsec functionality, including the reference identifier for the peer;
- Manage the TOE’s trust store and designate X.509v3 certificates as trust anchors;
- Ability to generate Certificate Signing Request (CSR) and process CA certificate response;
- Ability to administer the TOE locally;
- Ability to configure the local session inactivity time before session termination or locking;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- Ability to manage the trusted public keys database;
- Ability to configure firewall rules;
- Ability to define packet filtering rules and associate those rules to network interfaces and order them by priority.

6.11 Protection of the TSF

SFRs:	FPT_SKP_EXT.1, FPT_APW_EXT.1, FPT_TST_EXT.1, FPT_TST_EXT.3, FPT_TUD_EXT.1, FPT_STM_EXT.1, FPT_FLS.1/SelfTest

115 The TOE prevents the reading of all pre-shared keys, symmetric keys, and private keys stored within the TOE boundary.

116 Pre-shared keys, administrator passwords, and other credentials for the secure operation of the TOE are stored in the TOE's configuration file. Authorized administrators are allowed to enter this information through the communications paths such as the local console or HTTPS GUI. Once the password is entered, the TOE encrypts the password using AES-128 and writes the password to the configuration file permanently obscuring the contents. This configuration file containing encrypted passwords is available through the local console and HTTPS GUI by viewing a full configuration or backup of the configuration. The AES key for the protection of this configuration file and its passwords is generated by the TOE when the TOE is initialized and put into FIPS mode. There are no interfaces available to administrators, or otherwise enabled roles, that are designed specifically for the purpose of viewing keys and passwords in plaintext.

117 The TOE performs the following self-tests upon initialization:

a) CPU and Memory BIOS self-tests

   i) CPU and memory are initialized by exercising a set of known answer tests and the BIOS is compared against a known checksum of the image. The memory is zeroized and then has a random pattern written and read from the memory.

b) Boot loader image verification

   i) The boot loader integrity check is done through a cyclic redundancy check (CRC). If the CRC fails, the FortiGate unit will encounter an error during the boot process. Firmware images are signed, and the signature is attached to the code as it is built. When upgrading an image, the running OS will generate a signature and compare it with the signature attached to the image. If the signatures do not match, the new OS will not load.

c) Noise source tests

   i) The noise source is started, and pattern analysis is done on the output to ensure that the source is not stuck in a cryptographically weak state. These include both the repetition and adaptive proportion tests

d) FIPS 140-2 Cryptographic Known Answer Tests (KAT)

   i) Comparison of various cryptographic functions against an expected set of values to verify the correct operation of the cryptographic functions necessary to fulfill the SFRs.

   ii) This test can also be executed on demand at the discretion of the administrator.

118 The above tests ensure that the CPU and memory utilized by the TOE are functioning as intended, the BIOS and boot loader image are authentic and stable, the noise source used for entropy generation is functioning at capability and that the cryptographic algorithms used by the TOE are operating correctly. Together, these tests ensure that the TOE is operating at its intended level of capability.

119 The TOE will enter the FIPS Error Mode where all network interfaces are shut down and traffic is blocked if any of the following self-tests fail:

a) CPU and Memory BIOS tests

b) Boot loader image verification

c) Noise source tests

d) FIPS 140-2 Cryptographic KAT's

120 To resolve an error resulting from a test failure, a power cycle should be performed. When the device completes the boot up operation, this is evidence that the self-tests have passed, and the TOE including all cryptographic functions are operating correctly.

121 All cryptographic self-tests (including KATs) are performed in accordance with TSF-provided capabilities specified in FCS_COP.1/SigGen.

122 Additionally, the TOE may receive traffic above the capacity of the product it will drop all packets above this capacity. These events are logged to the audit log of the TOE.

123 The administrator may query the current version of the TOE via the GUI or CLI interfaces. The administrator also has the option to manually update the TOE.

124 Updates to the TOE are applied in accordance with the following process:

a) The administrator downloads the upgrade image/package from the Fortinet website.

b) Once downloaded, the administrator must transfer the image to the TOE via a trusted path (e.g., the web interface).

c) Upon initiating the update process, the TOE will attempt to verify the integrity and authenticity of the update package. This is achieved via the verification of a 2048-bit RSA signature that is applied to the package by the Fortinet development team.

d) If the signature cannot be verified, or the integrity of the package cannot be confirmed, the upgrade will fail, and an audit log generated accordingly.

e) If the signature is verified correctly and the integrity of the package is confirmed, the upgrade will be applied, and the TOE restarted.

125 The TOE supports the ability to synchronize with an NTP server or maintain its own time source which is free from outside interference. The TOE contains an internal battery-backed hardware clock for reliability. The Security Administrator can manually set the date and time during initial TOE configuration and may change the time during operation. The TOE makes use of time for the following purposes:

a) Generating audit logs (timestamps);

b) Session timeouts (lockout enforcement);

c) Cryptographic key regeneration intervals;

d) Certificate validation.

6.11.1 TOE Initialization

126 The Fortinet family of appliances provides a secure initialization procedure to ensure the integrity of the image and correct cryptographic functioning of the product prior to any information flowing. The product starts from a powered down state and no signals on the wire. The device then powers on and undergoes the following initialization process:

a) Bootstrap and Boot Loader

b) Verification of the kernel, firmware and software images via RSA 2048-bit signatures

c) Loading and Initialization of:

i) Kernel;

ii) Firmware;

iii) Cryptographic known answer tests;

iv) Entropy gathering and DRBG initialization; and

v) Cryptographic module.

127 Once the kernel, firmware and cryptographic services have been initialized the TOE loads the configured firewall rules. The configuration file is then consulted and are initialized and configured with their network settings as specified and if appropriate transitioned to the link up state. At this point packets may begin flowing through the various network interfaces. The CLI daemon is then started followed by the Web and the TOE is available for login to accept administrative connections.

6.12 TOE Access

SFRs:	FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4, FTA_TAB.1

128 TOE administrators may access the TOE remotely (via the HTTPS web GUI or SSH) or locally (via the console port).

129 The TOE permits administrators to define a session lifetime/inactive timer for both local and remote sessions. Once this time limit has been met, the TOE will automatically close the session (local or remote) that was inactive and require TOE administrators to re-authenticate before any access to TSF data is permitted. TOE administrators may also manually close their sessions. TOE administrators terminate their sessions via the Log Out button at the Web GUI and the exit command via the SSH and local CLI.

130 Users connecting to the TOE will be presented with a warning and consent banner prior to authentication.

6.13 Trusted Path/Channels

SFRs:	FTP_ITC.1, FTP_ITC.1/VPN, FTP_TRP.1/Admin

131 The TOE provides an Inter-TSF trusted channel between itself and the following entities:

a) Between the TOE and a FortiAnalyzer logging platform using TLS (initiated by the TOE); and

b) Between the TOE and VPN endpoints using IPsec (initiated by the TOE or endpoints).

132 Administrators may utilize an IPsec tunnel on top of SSH or HTTPS when performing remote administration to provide additional transport security.

133 The TOE provides a trusted path between itself and remote administrative users using the following protocols:

a) HTTPS (in compliance with RFC 2818) for the Web GUI; and

b) SSH in compliance with the following RFCs: 4251, 4252, 4253, 4254, 5647, 5656, 6668, 8268, 8308, and 8332.

134 These protocols implement cryptographic algorithms to provide data transport security and integrity, preventing unauthorized access to (or modification of) data sent between the TOE and remote administrative users.

6.14 Stateful Traffic/Packet Filtering

SFRs:	FFW_RUL_EXT.1, FPF_RUL_EXT.1

135 The TOE permits the configuration of stateful packet filtering policies. The following protocols and associated attributes are configurable within each policy:

a) ICMPv4 (RFC 792)

i) Type; and

ii) Code

b) ICMPv6 (RFC 4443)
   i) Type; and
   ii) Code
c) IPv4 (RFC 791)
   i) Source address;
   ii) Destination Address; and
   iii) Transport Layer Protocol
d) IPv6 (RFC 8200)
   i) Source address;
   ii) Destination Address;
   iii) Transport Layer Protocol
e) TCP (RFC 793)
   i) Source Port; and
   ii) Destination Port
f) UDP (RFC 768)
   i) Source Port; and
   ii) Destination Port

136 The TOE does not support UDP-Lite for IPv4 or IPv6.

137 Testing is performed by the developer during the development and QA stages to ensure conformance with each protocol RFC with changes being made as needed to ensure compliance in their implementations.

138 Rules can be configured to permit or drop traffic (with the generation of audit log entries for either option).

139 Each rule can be tied to a specific interface (port1, wan1, etc.).

140 Each packet that arrives on an interface is subject to the enforcement of stateful traffic filtering. This filtering verifies if the connection is part of an established session or if it is a new connection. If the security attributes of the incoming connection request match those already present for an entry in the state table of the TOE the information flow is automatically allowed. Otherwise, this is considered a new connection attempt.

141 For a new connection attempt a list of default rules, and then administrator-defined security rules are consulted in their sequence order until a match is found for that packet. The packet is then allowed, denied, or dropped based on the configuration of this rule.

142 The session database is consulted to see if an additional session can be created by examining how many currently exist in the database. If this number is below the hardware limit sessions are established by writing the attributes and a TTL into the session database. If the connection is allowed a new session is written into the list of established sessions and can be used to allow subsequent packets for this connection. If logging is enabled for the rule the audit event is sent in real time to the audit server.

143 Any new session will have the first packet of the exchange inspected according to the firewall table as described above, such as the TCP SYN packet during a typical TCP session negotiation for both the sender and receiver. The TOE will write to the session table the expected source and destination ports for this communication flow based on the observed IP headers.

144 For FTP, the initial handshake communication on port 21 for FTP will be inspected, as well as the server response indicating the expected data and control communication ports. A session will be written to the state table reflecting the expected source and destination ports based on this packet inspection.

145 The TOE utilizes a session database to track all active sessions using TCP, UDP and ICMP protocols. The following variables are utilized in the management of these sessions:

a) TCP:
   i) Source and Destination addresses
   ii) Source and Destination ports
   iii) Sequence number
   iv) Flags

b) UDP:
   i) Source and Destination addresses
   ii) Source and Destination ports

c) ICMP:
   i) Source and Destination addresses
   ii) Type
   iii) Code

146 Periodically old sessions exceeding their TTL are removed from the database. Sessions that have been closed are similarly removed from the database. UDP and ICMP are connectionless protocols that do not have connection or protocol states, therefore UDP and ICMP session timeouts determine how long UDP and ICMP session information is kept in the session table.

147 Each FortiGate™ appliance has a pre-defined number of sessions it can track and is specified on the specifications sheet.

148 When encountered by the TOE, the following packets will be automatically dropped and an audit log generated for each event:

a) Packets which are invalid fragments (see below);

b) Fragments that cannot be completely re-assembled;

c) Packets where the source address is defined as being on a broadcast network;

d) Packets where the source address is defined as being on a multicast network;

e) Packets where the source address is defined as being a loopback address;

f) 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) 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) Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified.

i) Packets where the source address is equal to the address of the network interface where the network packet was received;

j) Packets where the source or destination address of the network packet is a linklocal address; and

k) Packets where the source address does not belong to the networks associated with the network interface where the network packet was received - the TOE implements Reverse Path Forwarding (RPF), also called Anti Spoofing. This prevents an IP packet from being forwarded if its source IP address either does not belong to a locally attached subnet (local interface), or be a hop on the routing between the TOE and another source (static route, RIP, OSPF, BGP).

149 The TOE is capable of detecting fragmented packets. When fragmented packets arrive at their destination, they are reassembled and read. If the fragments do not arrive together, they must be held until all of the fragments arrive. Reassembly of a packet requires all of the fragments. The TOE in the evaluated configuration will attempt to reassemble fragmented packets. When these packets arrive at the TOE they will be held by the TOE for reassembly until the TTL expires. Should the TOE detect that there is a missing or invalid fragment (i.e. first fragment is too small, fragment offset is too small or fragment is out of bounds) during the reassembly the packet will be dropped and logged. IP integrity header checking reads the packets to verify if a packet is a valid TCP, UDP, ICMP, SCTP or GRE packet. Verification is also performed to ensure the protocol header is the correct length. This behavior is not capable of being modified or overwritten by the TOE administrator.

150 Incoming packets are inspected against the session database. Sessions that match all the security attributes and do not exceed the TTL are automatically passed on to their destination. Packets that do not match the attributes in the session database are then compared to the defined firewall rules for that interface identifier based on their unique numerical order. Packets that are permitted are passed to their destination, packets marked for logging are written to the audit log and packets marked for dropping are discarded.

151 Packet rules are enforced in the order defined by the administrator. If no matching rule is found, the TOE will automatically deny the packets and generate a log entry accordingly.

152 The TOE maintains half-open TCP sessions in the same manner as full TCP sessions. Once the administrator-defined limit for total sessions is met, sessions (both valid and half-open) are automatically closed based on their timeout value (if not cleared manually by an administrator).

153 All received network packets are processed by the TOE policy engine. The policy engine does stateful filtering of the received network packets according to the configured firewall policies. The TOE kernel monitors the state of any running processes, including the policy engine and VPN processes.

154 The network interfaces of the TOE remain down until the self-tests have passed and all processes are up and running. The failure of any of the self-tests during operation results in the network interfaces being downed and all traffic blocked. During operation, if any of the processes fail or terminate unexpectedly, the kernel will block traffic - i.e. the TOE fails closed.

155 The TOE also implements a conserve mode as a self-protection measure if a memory shortage occurs. Conserve mode activates protection measures in order to recover memory space such as throttling traffic. In extreme cases conserve mode will cause any new connection requests to be dropped. When sufficient memory is recovered to resume normal operation, the TOE exits conserve mode state and releases the protection measures.

7 Rationale

7.1 Conformance Claim Rationale

156 The following rationale is presented with regard to the PP/PP-Modules conformance claims:

a) **TOE type.** As identified in section 2.1, the TOE is firewall with VPN and packet filtering capabilities consistent with the claimed PP/PP-Modules.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are reproduced directly from the claimed PP/PP-Modules.

c) **Security objectives.** As shown in section 4, the security objectives are reproduced directly from the claimed PP/PP-Modules.

d) **Security requirements.** As shown in section 5, the security requirements are reproduced directly from the claimed PP/PP-Modules. No additional requirements have been specified.

7.2 Security Objectives Rationale

157 All security objectives are drawn directly from the claimed PP/PP-Modules.

7.3 Security Requirements Rationale

158 All security requirements are drawn directly from the claimed PP/PP-Modules in accordance with exact conformance. No consistent SFR rationale is presented in the claimed PP/PP-Modules, therefore no rationale is reproduced in this ST.

Annex A: Extended Components Definition

159 Refer to the Extended Components Definition of the Protection Profile and Protection Profile Modules claimed in Section 1.3.

Annex B: CAVP Certificates

Annex B.1: SFR Coverage

Table 23: CAVP SFR Coverage Mapping

SFR	Selections	Usage	CAVP	Notes
FCS_CKM.1 Cryptographic Key Generation	RSA KeyGen (186-5)	TLS, SSH	A2240 A2241 A2242	Fortinet FortiOS CP9 Cryptographic Library Fortinet FortiOS CP9Lite Cryptographic Library Fortinet FortiOS CP9XLite Cryptographic Library
	ECDSA KeyGen (186-5)	TLS, IPsec	A6643	Fortinet FortiOS SSL Cryptographic Library
	FFC – Safe Prime Groups	IPsec, TLS, SSH	CCTL-tested	
FCS_CKM.1/IKE Cryptographic Key Generation (for IKE Peer Authentication)	RSA KeyGen (186-5)	IPsec	A2240 A2241 A2242	Fortinet FortiOS CP9 Cryptographic Library Fortinet FortiOS CP9Lite Cryptographic Library Fortinet FortiOS CP9XLite Cryptographic Library
	ECDSA KeyGen (186-5)	IPsec	A6643	Fortinet FortiOS SSL Cryptographic Library
FCS_CKM.2 Cryptographic Key Establishment	ECC	TLS, SSH	A6643	Fortinet FortiOS SSL Cryptographic Library

SFR	Selections	Usage	CAVP	Notes
	KAS-ECC-SSC SP800-56Ar3 KDF IKEv1 KDF IKEv2 KDF TLS KDF SSH	IPsec	A6643	Fortinet FortiOS SSL Cryptographic Library
	FFC – Safe Prime Groups	IPsec, TLS, SSH	CCTL-tested	
FCS_COP.1/DataEncryption	AES-CBC-128/256	TLS	A2240 A2241 A2242	Fortinet FortiOS CP9 Cryptographic Library Fortinet FortiOS CP9Lite Cryptographic Library Fortinet FortiOS CP9XLite Cryptographic Library
	AES-CBC-256	IPsec	A2240 A2241 A2242	Fortinet FortiOS CP9 Cryptographic Library Fortinet FortiOS CP9Lite Cryptographic Library Fortinet FortiOS CP9XLite Cryptographic Library

SFR	Selections	Usage	CAVP	Notes
	AES-GCM-128/256	TLS, SSH	A2240 A2241 A2242	Fortinet FortiOS CP9 Cryptographic Library Fortinet FortiOS CP9Lite Cryptographic Library Fortinet FortiOS CP9XLite Cryptographic Library
	AES-GCM-256	IPsec	A2240 A2241 A2242	Fortinet FortiOS CP9 Cryptographic Library Fortinet FortiOS CP9Lite Cryptographic Library Fortinet FortiOS CP9XLite Cryptographic Library
FCS_COP.1/SigGen	RSA SigGen/SigVer (186-5)	IPsec, TLS, SSH, Trusted Update	A2240 A2241 A2242	Fortinet FortiOS CP9 Cryptographic Library Fortinet FortiOS CP9Lite Cryptographic Library Fortinet FortiOS CP9XLite Cryptographic Library
	ECDSA SigGen/SigVer (186-4) *P-256 only*	TLS, SSH, IPsec	A2240 A2241 A2242	Fortinet FortiOS CP9 Cryptographic Library Fortinet FortiOS CP9Lite Cryptographic Library Fortinet FortiOS CP9XLite Cryptographic Library
	ECDSA SigGen/SigVer (186-5) *P-384 & P-521*	TLS, SSH, IPsec	A6643	Fortinet FortiOS SSL Cryptographic Library
FCS_COP.1/Hash	SHA-1, SHA-256,	IPsec, Password Hashing	A6643	Fortinet FortiOS SSL Cryptographic Library

SFR	Selections	Usage	CAVP	Notes
	SHA-384, SHA-512	TLS, SSH	A6643	Fortinet FortiOS SSL Cryptographic Library
FCS_COP.1/KeyedHash	HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	IPsec	A6643	Fortinet FortiOS SSL Cryptographic Library
	HMAC-SHA-1 HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	TLS, SSH	A6643	Fortinet FortiOS SSL Cryptographic Library
FCS_RBG_EXT.1	CTR_DRBG (AES)	TOE RBG	A6641	Fortinet FortiOS FIPS Cryptographic Library

Annex B.2: CAVP Hardware Mapping

Refer to Table 4: TOE Hardware Models for CAVP Hardware mapping.