FortiGate/FortiOS 6.2.7

Security Target

Version 1.7

January 2022

Document prepared by

www.lightshipsec.com

Document History

Version	Date	Description
1.0	26 Apr 2021	Addressed issues raised by the scheme.
1.1	13 May 2021	Updated hardware models. Addressed additional issues.
1.2	28 June 2021	Updated guidance documentation and build number.
1.3	28 June 2021	Clarify vND use case.
1.4	08 July 2021	Update to MOD IPS
1.5	14 September 2021	Addressed CB OR.
1.6	30 November 2021	Removed FGT-100F/101F
1.7	14 January 2021	NIAP PCL updates

Table of Contents

1 Introduction 5
1.1 Overview 5
1.2 Identification 5
1.3 Conformance Claims 5
1.4 Terminology 6
2 TOE Description 8
2.1 Type 8
2.2 Usage 8
2.3 Logical Scope / Security Functions 9
2.4 Physical Scope 10
3 Security Problem Definition 19
3.1 Threats 19
3.2 Assumptions 23
3.3 Organizational Security Policies 25
4 Security Objectives 25
4.1 Security Objectives for the TOE 25
4.2 Security Objectives for the Environment 27
5 Security Requirements 30
5.1 Conventions 30
5.2 Extended Components Definition 30
5.3 Functional Requirements 30
5.4 Assurance Requirements 58
6 TOE Summary Specification 59
6.1 Security Audit 59
6.2 Cryptographic Support 59
6.3 HTTPS/TLS 64
6.4 SSH 65
6.5 IPsec 66
6.6 Residual Data Protection 67
6.7 Identification and Authentication 67
6.8 X509 Certificates 68
6.9 Security Management 69
6.10 Protection of the TSF 70
6.11 TOE Access 72
6.12 Trusted Path/Channels 72
6.13 Stateful Traffic/Packet Filtering 73
6.14 Intrusion Prevention (IPS) 75
7 Rationale 79
7.1 Conformance Claim Rationale 79
7.2 Security Objectives Rationale 79
7.3 Security Requirements Rationale 79
Annex A: Extended Components Definition 80
Annex B: CAVP Certificates 81
Annex B.1: SFR Coverage 81
Annex B.2: CAVP Libraries 86

Annex B.3: CAVP Hardware Mapping 90
Annex B.3: CAVP Virtual Appliance Coverage 97

List of Tables

Table 1: Evaluation identifiers 5
Table 2: NIAP Technical Decisions 5
Table 3: Terminology 6
Table 4: TOE Hardware Models 10
Table 5: TOE Virtual Appliance and Related Hardware 16
Table 6: Threats (CPP_ND) 19
Table 7: Threats (MOD_CPP_FW) 20
Table 8: Threats (MOD_VPNGW) 20
Table 9: Threats (MOD_IPS) 22
Table 10: Assumptions (CPP_ND) 23
Table 11: Assumptions (MOD_IPS) 24
Table 12: Assumptions (MOD_VPNGW) 25
Table 13: Organizational Security Policies (CPP_ND) 25
Table 14: Organizational Security Policies (MOD_IPS) 25
Table 15: Security Objectives for the TOE (MOD_CPP_FW) 25
Table 16: Security Objectives for the TOE (MOD_VPNGW) 26
Table 17: Security Objectives for the TOE (MOD_IPS) 27
Table 18: Security Objectives for the Environment (CPP_ND) 27
Table 19: Security Objectives for the Environment (MOD_CPP_FW) 29
Table 20: Security Objectives for the Environment (MOD_VPNGW) 29
Table 21: Security Objectives for the Environment (MOD_IPS) 29
Table 22: Summary of SFRs 30
Table 23: Assurance Requirements 58
Table 24: Key Generation Methods 59
Table 25: Key Establishment Methods 60
Table 26: Cryptographic Methods 60
Table 27: Keys and CSPs 62
Table 28: CAVP SFR Coverage Mapping 81
Table 29: CAVP Libraries & Capabilities Mapping 86
Table 30: CAVP Hardware Coverage 90
Table 31: TOE Virtual Appliance and related Hardware 97

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the Fortinet FortiGate/FortiOS 6.2.7 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 FortiGate next-generation firewall (NGFW) appliances running FortiOS software provide high performance, multilayered validated security and granular visibility for end-to-end protection across the entire enterprise.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	FortiGate/FortiOS 6.2.7 Version 6.2.7 Build 5067
Security Target	FortiGate/FortiOS 6.2.7 Security Target, v1.7

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, Intrusion Prevention Systems (IPS), Stateful Traffic Filter Firewalls, and Virtual Private Network (VPN) Gateways, v1.0

i) Base-PP: collaborative Protection Profile for Network Devices, v2.2e (CPP_ND)

ii) PP-Module: PP-Module for Stateful Traffic Filter Firewalls, v1.4e (MOD_CPP_FW)

iii) PP-Module for Virtual Private Network (VPN) Gateways, v1.1 (MOD_VPNGW)

iv) PP-Module for Intrusion Prevention Systems, v1.0 (MOD_ IPS)

e) NIAP Technical Decisions per Table 2

Table 2: NIAP Technical Decisions

TD #	Name
TD0527	Updates to Certificate Revocation Testing (FIA_X509_EXT.1)
TD0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4
TD0536	NIT Technical Decision for Update Verification Inconsistency
TD0537	NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3
TD0538	NIT Technical Decision for Outdated link to allowed-with list

TD #	Name
TD0545	NIT Technical Decision for Conflicting FW rules cannot be configured (extension of Rfl#201837)
TD0546	NIT Technical Decision for DTLS - clarification of Application Note 63
TD0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN
TD0549	Consistency of Security Problem Definition update for MOD_VPNGW_v1.0 and MOD_VPNGW_v1.1
TD0551	NIT Technical Decision for Incomplete Mappings of OEs in FW Module v1.4+Errata
TD0555	NIT Technical Decision for RFC Reference incorrect in TLSS Test
TD0556	NIT Technical Decision for RFC 5077 question
TD0563	NiT Technical Decision for Clarification of audit date information
TD0564	NiT Technical Decision for Vulnerability Analysis Search Criteria
TD0569	NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7
TD0570	NiT Technical Decision for Clarification about FIA_AFL.1
TD0571	NiT Technical Decision for Guidance on how to handle FIA_AFL.1
TD0572	NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers
TD0580	NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e
TD0581	NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3
TD0590	Mapping of operational environment objectives
TD0591	NIT Technical Decision for Virtual TOEs and hypervisors
TD0592	NIT Technical Decision for Local Storage of Audit Records
TD0595	Administrative corrections to IPS PP-Module
TD0597	VPN GW IPv6 Protocol Support

1.4 Terminology

Table 3: Terminology

Term	Definition
BGP	Border Gateway Protocol

Term	Definition
CC	Common Criteria
CLI	Command Line Interface
cPP	Collaborative Protection Profile
EAL	Evaluation Assurance Level
EP	Extended Package
FW	Firewall
FortiGate	Fortinet NGFW hardware appliance(s)
FortiOS	Fortinet NGFW operating system
GUI	Graphical User Interface
IPS	Intrusion Prevention System
NDcPP	collaborative Protection Profile for Network Devices
NGFW	Next-Generation Firewall
OSPF	Open Shortest Path First
PP	Protection Profile
RIP	Routing Information Protocol
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality
UTM	Unified Threat Management
VPN	Virtual Private Network

2 TOE Description

2.1 Type

4 The TOE is a firewall that includes Virtual Private Network (VPN) and Intrusion Prevention System (IPS) capabilities. Industry terms for this TOE type include Next-Generation Firewall (NGFW) and Unified Threat Management (UTM).

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.

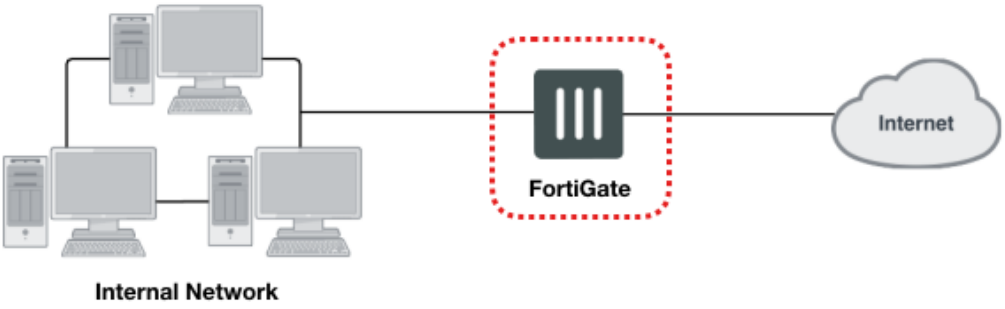


Figure 1: Example TOE deployment

2.2.2 Interfaces

6 The TOE interfaces are shown in Figure 2.

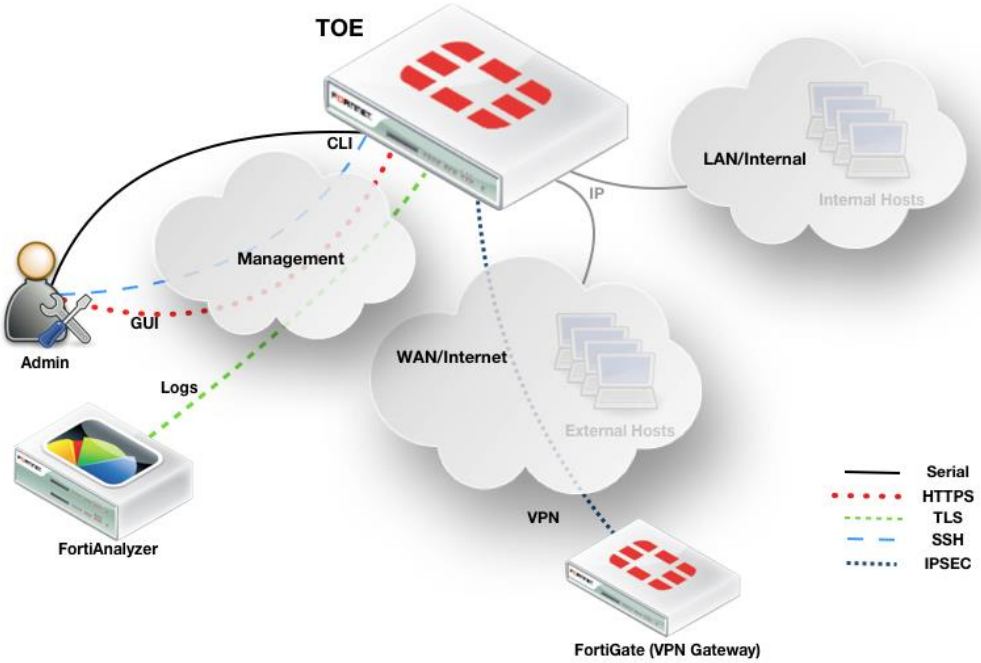


Figure 2: TOE interfaces

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

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

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

2.3 Logical Scope / Security Functions

8 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.

c) **Residual Data Protection.** The TOE ensures that data cannot be recovered once deallocated.

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.

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.

g) **TOE Access.** The TOE provides session management functions for local and remote administrative sections.

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.

j) **Intrusion Prevention.** The TOE allows for the enforcement of pre-defined or custom attack signatures, as part of a comprehensive intrusion prevention suite.

2.3.1 Functions not included in the TOE Evaluation

9 The FortiGate appliances are capable of a variety of functions and configurations which are not covered by the FWcPP and associated EPs.

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

a) High-Availability

b) FortiExplorer client

c) Anti-spam

d) Anti-virus
e) Content filtering
f) Web filtering
g) Use of syslog
h) FortiToken and FortiSSO Authentication
i) Stream Control Transmission Protocol (SCTP), BGP, RIP and DHCP protocols
j) Usage of the boot-time configuration menu to upgrade the TOE
k) Policy-based VPN
l) SSL VPN
m) Virtual domains (vdoms)
n) Logging to FortiCloud

2.4 Physical Scope

11 The physical boundary of the TOE includes the FortiGate hardware models shown in Table 4 and the virtual appliances and related hardware shown in Table 5 running FortiOS software identified in Table 1. The virtual appliances are evaluated as virtual Network Devices (vND), which is case 1 of Section 1.2 of NDcPP v2.2e. The TOE is shipped to the customer via commercial courier. The virtual appliances' deployment packages can be downloaded from the Customer Service & Support site.

Table 4: TOE Hardware Models

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy
FG-30E	Fortinet SoC3	ARMv7-A	1 GB	128MB	n/a	n/a	Token
FWF-30E	Fortinet SoC3	ARMv7-A	1 GB	128MB	n/a	n/a	Token
FG-40F	Fortinet SoC4	ARMv8	2GB	4GB	n/a	CP9Lite	CP9Lite
FG-50E	Fortinet SoC3	ARMv7-A	2 GB	128MB	n/a	n/a	Token
FWF-50E	Fortinet SoC3	ARMv7-A	2 GB	128MB	n/a	n/a	Token
FG-51E	Fortinet SoC3	ARMv7-A	2 GB	128MB	32GB	n/a	Token
FWF-51E	Fortinet SoC3	ARMv7-A	2 GB	128MB	32GB	n/a	Token
FG-52E	Fortinet SoC3	ARMv7-A	2 GB	128MB	32GB x2 (64GB)	n/a	Token

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy
FG-60E	Fortinet SoC3	ARMv7-A	2 GB	8GB	n/a	CP9Lite	SoC3
FG-60E-DSL	Fortinet SoC3	ARMv7-A	2 GB	8GB	n/a	CP9Lite	SoC3
FG-60E-PoE	Fortinet SoC3	ARMv7-A	2 GB	8GB	n/a	CP9Lite	SoC3
FG-60F	Fortinet SoC4	ARMv8	2 GB	8GB	n/a	CP9Lite	CP9Lite
FWF-60E	Fortinet SoC3	ARMv7-A	2 GB	8GB	n/a	CP9Lite	SoC3
FWF-60E-DSL	Fortinet SoC3	ARMv7-A	2 GB	8GB	n/a	CP9Lite	SoC3
FWF-60F	Fortinet SoC4	ARMv8	2 GB	8GB	n/a	CP9Lite	CP9Lite
FG-61E	Fortinet SoC3	ARMv7-A	2 GB	8GB	128GB	CP9Lite	SoC3
FG-61F	Fortinet SoC4	ARMv8	2 GB	8GB	128GB	CP9Lite	CP9Lite
FWF-61E	Fortinet SoC3	ARMv7-A	2 GB	8GB	128GB	CP9Lite	SoC3
FWF-61F	Fortinet SoC4	ARMv8	2 GB	8GB	128GB	CP9Lite	CP9Lite
FG-80E	Fortinet SoC3	ARMv7-A	2 GB	8GB	n/a	CP9Lite	SoC3
FG-80E-PoE	Fortinet SoC3	ARMv7-A	2 GB	8GB	n/a	CP9Lite	SoC3
FG-81E	Fortinet SoC3	ARMv7-A	2 GB	8GB	128GB	CP9Lite	SoC3
FG-81E-PoE	Fortinet SoC3	ARMv7-A	2 GB	8GB	128GB	CP9Lite	SoC3
FG-100E	Fortinet SoC3	ARMv7-A	4 GB	8GB	n/a	CP9Lite	SoC3
FG-100EF	Fortinet SoC3	ARMv7-A	4 GB	8GB	n/a	CP9Lite	SoC3

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy
FG-100F	Fortinet SoC4	ARMv8	4 GB	8GB	n/a	CP9Lite	CP9Lite
FG-101E	Fortinet SoC3	ARMv7-A	4 GB	8GB	480GB	CP9Lite	SoC3
FG-101F	Fortinet SoC4	ARMv8	4 GB	8GB	480GB	CP9Lite	CP9Lite
FG-140E	Fortinet SoC3	ARMv7-A	4 GB	8GB	n/a	CP9Lite	SoC3
FG-140E-PoE	Fortinet SoC3	ARMv7-A	4 GB	8GB	n/a	CP9Lite	SoC3
FG-200E	Intel Celeron G1820	Haswell	4GB	16GB	n/a	CP9	CP9
FG-201E	Intel Celeron G1820	Haswell	4GB	16GB	480GB	CP9	CP9
FG-300D	Intel i3-3220	Ivy Bridge	8 GB	16GB	120GB	CP8	Token
FG-300E	Intel i5-6500	Skylake	8GB	16GB	n/a	CP9	CP9
FG-301E	Intel i5-6500	Skylake	8GB	16GB	n/a	CP9	CP9
FG-400D	Intel i3-3220	Ivy Bridge	8 GB	16GB	n/a	CP8	Token
FG-400E	Intel i5-8500	Coffeelake	8 GB	16GB	n/a	CP9	CP9
FG-401E	Intel i5-8500	Coffeelake	8 GB	16GB	480GB	CP9	CP9
FG-500D	Intel Xeon E3-1225v2	Ivy Bridge	8 GB	16GB	120GB MLC	CP8	Token
FG-500E	Intel i7-6700	Skylake	16GB	16GB	n/a	CP9	CP9
FG-501E	Intel i7-6700	Skylake	16GB	16GB	n/a	CP9	CP9

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy
FG-600D	Intel i7-3770	Ivy Bridge	8 GB	16GB	120GB	CP8	Token
FG-600E	Intel i7-8700	Coffeelake	16 GB	16GB	n/a	CP9	CP9
FG-601E	Intel i7-8700	Coffeelake	16 GB	16GB	480GB	CP9	CP9
FG-900D	Intel Xeon E3-1225v3	Haswell	16 GB	2GB	256GB	CP8	Token
FG-1000D	Intel Xeon E5-1275v3	Haswell	16 GB	4GB	256GB	CP8	Token
FG-1100E	Intel Xeon E-2186G	Coffeelake	16 GB	16GB	n/a	CP9	CP9
FG-1101E	Intel Xeon E-2186G	Coffeelake	16 GB	16GB	960GB	CP9	CP9
FG-1200D	Intel Xeon E5-1275v3	Haswell	16 GB	16GB	240GB	CP8	Token
FG-1500D	Intel Xeon E5-1650v2	Ivy Bridge	16 GB	32GB	2x 240GB (480GB)	CP8	Token
FG-1500D-DC	Intel Xeon E5-1650	Ivy Bridge	16GB	32GB	2x 240GB (480GB)	n/a	Token
FG-1500DT	Intel Xeon E5-1650v2	Ivy Bridge	16GB	32GB	2x 240GB/ 480GB	n/a	Token
FG-2000E	Intel Xeon E5-1660v4	Broadwell	32 GB	16GB	480GB	CP9	CP9
FG-2200E	Intel Xeon Gold 6126	Purely	24 GB	16GB	n/a	CP9	CP9
FG-2201E	Intel Xeon Gold 6126	Purley	24 GB	16GB	2TB	CP9	CP9

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy
FG-2500E	Intel Xeon E5-1650v3	Broadwell	32 GB	16GB	480GB	CP9	CP9
FG-3000D	Intel Xeon E5-2650v3	Haswell	64 GB	16GB	480GB	CP8	Token
FG-3100D	Intel Xeon E5-2660v3	Haswell	64 GB	16GB	480GB	CP8	Token
FG-3100D-DC	Intel Xeon E5-2660 v3	Haswell	64GB	16GB	480GB	CP8	Token
FG-3200D-DC	Intel Xeon E5-2670 v3	Haswell	64GB	16GB	2x 480GB (960GB)	CP8	Token
FG-3200D	Intel Xeon E5-2670v3	Haswell	64 GB	16GB	2x 480GB (960GB)	CP8	Token
FG-3300E	Intel Xeon Gold 5118	Purely	96 GB	16GB	n/a	CP9	CP9
FG-3301E	Intel Xeon Gold 5118	Purley	96 GB	16GB	2TB	CP9	CP9
FG-3400E	Intel Xeon Gold 6130	Purely	96 GB	16GB	n/a	CP9	CP9
FG-3401E	Intel Xeon Gold 6130	Purley	96 GB	16GB	2TB	CP9	CP9
FG-3600E	Intel Xeon Gold 6152	Purley	96 GB	16GB	n/a	CP9	CP9
FG-3601E	Intel Xeon Gold 6152	Purely	96 GB	16GB	2TB	CP9	CP9
FG-3700D	Intel Xeon E5-2680V2	Ivy Bridge	64 GB	16GB	2x 480GB (960GB)	CP8	Token

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy
FG-3800D	Intel Xeon E5-2680V2	Ivy Bridge	64 GB	16GB	2x 480GB (960GB)	CP8	Token
FG-3810D	Intel Xeon E5-2680V2	Ivy Bridge	64 GB	16GB	2x 480GB (960GB)	CP8	Token
FG-3815D	Intel Xeon E5-2680V2	Ivy Bridge	64 GB	16GB	2x 480GB (960GB)	CP8	Token
FG-3960E	Intel Xeon E5-2650V4	Broadwell	256GB	16GB	n/a	CP9	CP9
FG-3980E	Intel Xeon E5-2680V4	Broadwell	256GB	16GB	n/a	CP9	CP9
FG-5001D*	Intel Xeon E5-2658V2	Ivy Bridge	32GB	16GB	240GB	CP8	Token
FG-5001E*	Intel Xeon E5-2690v4	Broadwell	64 GB	16GB	n/a	CP9	CP9
FG-5001E1	Intel Xeon E5-2690v4	Broadwell	64 GB	16GB	480GB	CP9	CP9
FGR-30D	Fortinet SoC3	ARMv7-A	1GB	128MB	n/a	n/a	Token
FGR-60F	Fortinet SoC4	ARMv8	2GB	8GB	n/a	CP9XLite	SoC4
FGR-60F-3G4G	Fortinet SoC4	ARMv8	2GB	8GB	n/a	CP9XLite	SoC4
FG-40F-3G4G	Fortinet SoC4	ARMv8	2 GB	4 GB	n/a	CP9XLite	SoC4
FG-60E-DSLJ	Fortinet SoC3	ARMv7-A	2 GB	8 GB	n/a	CP9Lite	SoC3
FG-60F	Fortinet SoC4	ARMv8	2 GB	8 GB	n/a	CP9XLite	SoC4

Model	CPU	Architecture	RAM	Boot	Storage	ASIC	Entropy
FG-61F	Fortinet SoC4	ARMv8	2 GB	8 GB	128 GB	CP9XLite	SoC4
FG-800D	Intel Xeon E3-1225 v3	Haswell	8 GB	16 GB	240 GB	CP8	Token
FWF-40F	Fortinet SoC4	ARMv8	2 GB	4 GB	n/a	CP9XLite	SoC4
FWF-40F-3G4G	Fortinet SoC4	ARMv8	2 GB	4 GB	n/a	CP9XLite	SoC4
FWF-50E-2R	Fortinet SoC3	ARMv7-A	2 GB	128 MB	n/a	n/a	Token
FWF-60E-DSLJ	Fortinet SoC3	ARMv7-A	2 GB	8GB	n/a	CP9Lite	SoC3
FWF-60F	Fortinet SoC4	ARMv8	2 GB	8 GB	n/a	CP9XLite	SoC4
FWF-61F	Fortinet SoC4	ARMv8	2 GB	8 GB	128 GB	CP9XLite	SoC4

* Blade mounted in FortiGate 5144C rack mount ATCA chassis.

Table 5: TOE Virtual Appliance and Related Hardware

Model	Description	Tested Hypervisor	Tested CPUs*	Entropy
FortiGate-VM01	1x vCPU core and unlimited RAM	VMware ESXi 6.7	Intel Xeon D-15xx Intel Xeon E3 15xx Intel Xeon E-22xx Intel Core i7-68xx	Token via USB pass-through
FortiGate-VM02	2x vCPU cores and unlimited RAM			
FortiGate-VM04	4x vCPU cores and unlimited RAM			
FortiGate-VM08	8x vCPU cores and unlimited RAM			
FortiGate-VM16	16x vCPU cores and unlimited RAM			
FortiGate-VM32	32x vCPU cores and unlimited RAM			
FortiGate-VMUL	Unlimited vCPU cores and RAM			

*Provided with PacStar 451/455 and MilDef CS9121

2.4.1 Guidance Documents

12 The TOE includes the following guidance documents (PDF/HTML):

a) FortiOS – Cookbook, Version 6.2.7, 01-627-538742-20210210

b) FortiOS 6.2 and FortiGate NGFW Appliances FIPS 140-2 and Common Criteria Technote, 01-627-672110-20210525

c) FortiOS 6.2.7 CLI Reference, 01-627-685877-20210209

d) FortiOS - Log Reference, Version 6.2.7, 01-627-538745-20201217

e) FortiOS - Hardening your FortiGate, Version 6.2.0, 01-620-554155-20201210

f) Custom IPS and Application Control Signature 3.6 Syntax Guide, 43-360-453749-20200225

g) Virtualization Guides:

  i) FortiOS 6.2 Virtualization Reference, https://docs.fortinet.com/vm

  ii) FortiGate virtual appliances documentation, https://docs.fortinet.com/document/fortigate-private-cloud/6.2.0/vmware-esxi-administration-guide/706376/about-fortigate-vm-on-vmware-esxi

h) Hardware Guides:

  i) FortiGate/FortiWiFi 30E/50E/51E 01-540-269598-20180808

  ii) FortiGate 52E 01-540-300075-20170907

  iii) FortiGate 60E/61E Series 01-540-367071-20181107

  iv) FortiGate 60E-DSL 01-560-442605-20200519

  v) QuickStart Guide FortiGate/FortiWiFi 40F/40F-3G4G & 60F/61F Series, August 4, 2020

  vi) FortiGate 80E/81E 01-543-402959-20180808

  vii) FortiGate 80E/81E-POE 01-542-391830-20180314

  viii) FortiGate 100E/101E 01-540-366134-20170913

  ix) FortiGate 100EF 01-543-403497-20170907

  x) FortiGate 140E/140E-PoE Series 01-543-404092-20180807

  xi) FortiGate 200E/201E 01-542-381079-20190912

  xii) FortiGate 300D 01-506-238488-20170824

  xiii) FortiGate 300E/301E 01-560-440261-20191010

  xiv) FortiGate 400D 01-523-277788-20170824

  xv) FortiGate 400E/401E 01-563-522532-20200427

  xvi) FortiGate 500D 01-523-278008-20190408

  xvii) FortiGate 500E/501E 01-560-440260-20191009

  xviii) FortiGate 600D 01-523-278008-20170907

  xix) FortiGate 600E/601E 01-602-519311-20190726

  xx) FortiGate 800D 01-540-273916-20200724

  xxi) FortiGate 900D 01-523-279315-20200406

  xxii) FortiGate 1000D 01-503-237227-20200406

  xxiii) FortiGate 1100E/1101E 01-620-24051-20190425

  xxiv) FortiGate 1200D 01-540-306494-20190613

xxv) FortiGate 2200E/2201E 01-600-231503-20200430
xxvi) FortiGate 3000D 01-522-266144-20170907
xxvii) FortiGate 3100D 01-5011-275737-20180711
xxviii) FortiGate 3200D 01-522-256537-20190321
xxix) FortiGate 1500D 01-523-211767-20200406
xxx) FortiGate 3300E/3301E 01-600-511354-20200430
xxxi) FortiGate 3400E/3401E Series 01-602-511354-20200225
xxxii) FortiGate 3600E/3601E Series 01-602-510285-20200225
xxxiii) FortiGate 3700D 01-540-292415-20190501-M
xxxiv) FortiGate 3800D 01-540-292415-20190828-M
xxxv) FortiGate 3810D 01-522-261444-20170901-M
xxxvi) FortiGate 3815D 01-540-292419-20170901-M
xxxvii) FortiGate-5001D 01-560-0242101-20170728
xxxviii) FortiGate 2000E/2500E 01-540-306896 -20170907
xxxix) FortiGate 3960E/3980E 01-540-376285-20180807
xl) FortiGate-5001E System Guide 01-600-410512-20190709
xli) FortiGate Rugged 30D Information 01-540-297736-20200812
xlii) FortiGate Rugged 60F/3G4G Series October 15, 2020
xliii) FortiGate 100F/101F Series QuickStart Guide December 11, 2020
xliv) FortiGate 1500DT QSG Supplement 01-540-297032-20200728
xlv) FortiGate 1500D QSG Supplement 01-523-211767-20200728
xlvi) FortiGate/FortiWiFi 40F & 60F Series QuickStart Guide August 4, 2020

13 Guides are available at: https://docs.fortinet.com/fortigate

2.4.2 Non-TOE Components

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

a) **Audit Server.** The TOE makes use of a FortiAnalyzer for remote logging.
b) **VPN Endpoints.** The TOE supports FortiGate VPN endpoints.
c) **CRL Web Server.** Web server capable of serving up CRLs over HTTP.

3 Security Problem Definition

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

3.1 Threats

Table 6: 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 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

Identifier	Description
	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.
T.PASSWORD_ CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other Network Devices.
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 7: 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 8: 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, which 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

Identifier	Description
	those external devices then the data contained within 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 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

Identifier	Description
	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 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.

Table 9: Threats (MOD_IPS)

Identifier	Description
T.NETWORK_DISCLOSURE	Sensitive information on a protected network might be disclosed resulting from ingress- or egress-based actions.
T. NETWORK_ACCESS	Unauthorized access may be achieved to services on a protected network from outside that network, or alternately services outside a protected network from inside the protected network. If malicious

Identifier	Description
	external devices are able to communicate with devices on the protected network via a backdoor then those devices may be susceptible to the unauthorized disclosure of information.
T.NETWORK_MISUSE	Access to services made available by a protected network might be used counter to operational environment policies. Devices located outside the protected network may attempt to conduct inappropriate activities while communicating with allowed public services. (e.g. manipulation of resident tools, SQL injection, phishing, forced resets, malicious zip files, disguised executables, privilege escalation tools and botnets).
T.NETWORK_DOS	Attacks against services inside a protected network, or indirectly by virtue of access to malicious agents from within a protected network, might lead to denial of services otherwise available within a protected network.

3.2 Assumptions

Table 10: 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). In the case of vNDs, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of the distributed TOE run inside more than one virtual machine (VM) on a single VS. There are no other guest VMs on the physical platform providing non-Network Device functionality.
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).

Identifier	Description
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.
A.VS_TRUSTED_ ADMINISTRATOR (applies to vNDs only)	The Security Administrators for the VS are assumed to be trusted and to act in the best interest of security for the organization. This includes not interfering with the correct operation of the device. The Network Device is not expected to be capable of defending against a malicious VS Administrator that actively works to bypass or compromise the security of the device.
A.VS_REGULAR_ UPDATES (applies to vNDs only)	The VS software is assumed to be updated by the VS Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.VS_ISOLATION (applies to vNDs only)	For vNDs, it is assumed that the VS provides, and is configured to provide sufficient isolation between software running in VMs on the same physical platform. Furthermore, it is assumed that the VS adequately protects itself from software running inside VMs on the same physical platform.
A.VS_CORRECT_ CONFIGURATION (applies to vNDs only)	For vNDs, it is assumed that the VS and VMs are correctly configured to support ND functionality implemented in VMs.

Table 11: Assumptions (MOD_IPS)

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.

Table 12: Assumptions (MOD_VPNGW)

Identifier	Description
A.CONNECTIONS	This assumption defines the TOE's placement in a network such that it is able to perform its required security functionality. The Base-PP does not define any assumptions about the TOE's architectural deployment so there is no conflict here.

3.3 Organizational Security Policies

Table 13: 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 users consent by accessing the TOE.

Table 14: Organizational Security Policies (MOD_IPS)

Identifier	Description
P.ANALYZE	Analytical processes and information to derive conclusions about potential intrusions must be applied to IPS data and appropriate response actions taken.

4 Security Objectives

4.1 Security Objectives for the TOE

Table 15: 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.

Identifier	Description
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 16: 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.
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. 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.

Identifier	Description
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	Compliant 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.

Table 17: Security Objectives for the TOE (MOD_IPS)

Identifier	Description
O.SYSTEM_MONITORING	To be able to analyze and react to potential network policy violations, the IPS must be able to collect and store essential data elements of network traffic on monitored networks.
O.IPS_ANALYZE	Entities that reside on or communicate across monitored networks must have network activity effectively analyzed for potential violations of approved network usage. The TOE must be able to effectively analyze data collected from monitored networks to reduce the risk of unauthorized disclosure of information, inappropriate access to services, and misuse of network resources.
O.IPS_REACT	The TOE must be able to react in real-time as configured by the Security Administrator to terminate and block traffic flows that have been determined to violate administrator-defined IPS policies.
O.TOE_ADMINISTRATION	To address the threat of unauthorized administrator access that is defined in the Base-PP, conformant TOEs will provide the functions necessary for an administrator to configure the IPS capabilities of the TOE.

4.2 Security Objectives for the Environment

Table 18: 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.

Identifier	Description
OE.NO_GENERAL_ PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.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.
OE.VM_ CONFIGURATION (applies to vNDs only)	For vNDs, the Security Administrator ensures that the VS and VMs are configured to • reduce the attack surface of VMs as much as possible while supporting ND functionality (e.g., remove unnecessary virtual hardware, turn off unused inter-VM communications mechanisms), and • correctly implement ND functionality (e.g., ensure virtual networking is properly configured to support network traffic, management channels, and audit reporting). The VS should be operated in a manner that reduces the likelihood that vND operations are adversely affected by virtualisation features such as cloning, save/restore, suspend/resume, and live migration. If possible, the VS should be configured to make use of features that leverage the VS's privileged position to provide additional security functionality. Such features could include malware detection through VM introspection, measured VM boot, or VM snapshot for forensic analysis.

Table 19: Security Objectives for the Environment (MOD_CPP_FW)

Identifier	Description
n/a	All objectives for the Operational Environment of the Base-PP apply also to this PP-Module. OE.NO_THRU_TRAFFIC_PROTECTION is still operative, but only for the interfaces in the TOE that are defined by the Base-PP and not the PP-Module.

Table 20: 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.

Table 21: Security Objectives for the Environment (MOD_IPS)

Identifier	Description
OE.CONNECTIONS	TOE administrators will ensure that the TOE is installed in a manner that will allow the TOE to effectively enforce its policies on network traffic of monitored networks.

5 Security Requirements

5.1 Conventions

16 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

17 Refer to Annex A: Extended Components Definition.

5.3 Functional Requirements

Table 22: Summary of SFRs

Requirement	Title	Source
FAU_GEN.1	Audit data generation	CPP_ND MOD_CPP_FW MOD_VPNGW
FAU_GEN.1/IPS	Audit data generation (IPS)	IPS_EP
FAU_GEN.2	User identity association	CPP_ND
FAU_STG_EXT.1	Security audit event storage	CPP_ND
FCS_CKM.1	Cryptographic key generation	CPP_ND
FCS_CKM.1/IKE	Cryptographic key generation (for IKE peer authentication)	MOD_VPNGW
FCS_CKM.2	Cryptographic key establishment	CPP_ND
FCS_CKM.4	Cryptographic key destruction	CPP_ND
FCS_COP.1/DataEncryption	Cryptographic operation (AES data encryption/decryption)	CPP_ND MOD_VPNGW
FCS_COP.1/SigGen	Cryptographic operation (Signature generation and verification)	CPP_ND
FCS_COP.1/Hash	Cryptographic operation (Hash algorithm)	CPP_ND
FCS_COP.1/KeyedHash	Cryptographic operation (Keyed hash algorithm)	CPP_ND

Requirement	Title	Source
FCS_RBG_EXT.1	Random bit generation	CPP_ND
FCS_HTTPS_EXT.1	HTTPS protocol	CPP_ND
FCS_SSHS_EXT.1	SSH server protocol	CPP_ND
FCS_TLSC_EXT.1	TLS Client protocol	CPP_ND
FCS_TLSC_EXT.2	TLS Client Support for Mutual Authentication	CPP_ND
FCS_TLSS_EXT.1	TLS Server protocol	CPP_ND
FCS_IPSEC_EXT.1	IPsec protocol	CPP_ND MOD_VPNGW
FDP_RIP.2	Full residual information protection	MOD_CPP_FW
FIA_AFL.1	Authentication failure handling	CPP_ND
FIA_PMG_EXT.1	Password management	CPP_ND
FIA_PSK_EXT.1	Pre-shared key composition	MOD_VPNGW
FIA_UAU_EXT.2	Password-based authentication mechanism	CPP_ND
FIA_UAU.7	Protected authentication feedback	CPP_ND
FIA_UIA_EXT.1	User identification and authentication	CPP_ND
FIA_X509_EXT.1/Rev	X.509 certificate validation	CPP_ND
FIA_X509_EXT.2	X.509 certificate authentication	CPP_ND MOD_VPNGW
FIA_X509_EXT.3	X.509 certificate requests	CPP_ND
FMT_MOF.1/ManualUpdate	Management of security functions behaviour (Trusted Update)	CPP_ND
FMT_MOF.1/Functions	Management of security functions behaviour	CPP_ND
FMT_MOF.1.1/Services	Management of security functions behaviour	CPP_ND
FMT_MTD.1/CoreData	Management of TSF data	CPP_ND
FMT_MTD.1/CryptoKeys	Management of TSF data	CPP_ND MOD_VPNGW
FMT_SMF.1	Specification of management functions	CPP_ND

Requirement	Title	Source
FMT_SMF.1/FFW	Specification of Management Functions	MOD_CPP_FW
FMT_SMF.1/IPS	Specification of management functions (IPS)	IPS_EP
FMT_SMF.1/VPN	Specification of Management Functions (VPN Gateway)	MOD_VPNGW
FMT_SMR.2	Restrictions on security roles	CPP_ND
FPT_APW_EXT.1	Protection of administrator passwords	CPP_ND
FPT_FLS.1/SelfTest	Fail secure (Self-test failures)	MOD_VPNGW
FPT_TST_EXT.1	TSF testing	CPP_ND MOD_VPNGW
FPT_TST_EXT.3	TSF Self-Test with Defined Methods	MOD_VPNGW
FPT_SKP_EXT.1	Protection of TSF data (for reading of all pre-shared, symmetric and private keys)	CPP_ND
FPT_TUD_EXT.1	Trusted updates	CPP_ND MOD_VPNGW
FPT_STM_EXT.1	Reliable time stamps	CPP_ND
FTA_SSL_EXT.1	TSF-initiated session locking	CPP_ND
FTA_SSL.3	TSF-initiated termination	CPP_ND
FTA_SSL.4	User-initiated termination	CPP_ND
FTA_TAB.1	Default TOE access banners	CPP_ND
FTP_ITC.1	Inter-TSF Trusted Channel	CPP_ND
FTP_ITC.1/VPN	Inter-TSF Trusted Channel (VPN Communications)	MOD_VPNGW
FTP_TRP.1/Admin	Trusted path	CPP_ND
FFW_RUL_EXT.1	Stateful traffic filtering	MOD_CPP_FW
FPF_RUL_EXT.1	Rules for Packet Filtering	MOD_VPNGW
IPS_ABD_EXT.1	Anomaly-based IPS functionality	IPS_EP
IPS_IPB_EXT.1	IP blocking	IPS_EP
IPS_NTA_EXT.1	Network traffic analysis	IPS_EP

Requirement	Title	Source
IPS_SBD_EXT.1	Signature-based IPS functionality	IPS_EP

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

d) *Specifically defined auditable events listed in ~~Table 2~~* ***the table below****.*

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
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 a IPsec SA.	Reason for failure
FCS_IPSEC_EXT.1	Session Establishment with peer	Entire packet contents of packets transmitted/received during session establishment
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS session	Reason for failure
FCS_TLSC_EXT.2	Failure to establish a TLS Session	Reason for failure
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 mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate	Reason for failure of certificate validation
	Any addition, replacement or removal of trust anchors in the TOE's trust store	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.

Requirement	Auditable Events	Additional Audit Record Contents
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.
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.	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	The termination of a local session by the session locking mechanism.	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.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.

Requirement	Auditable Events	Additional Audit Record Contents
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
FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol

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

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

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

FAU_GEN.1/IPS Audit data generation (IPS)

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

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

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

c) [All dissimilar IPS events;

d) All dissimilar IPS reactions;

e) Totals of similar events occurring within a specified time period;

f) Totals of similar reactions occurring within a specified time period;

g) *The events in the IPS table;*

h) *[no other auditable events]*].

Application note: Altered by TD0595.

FAU_GEN.1.2/IPS The TSF shall record within each **IPS auditable event** record at least the following information:

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

b) For each **IPS auditable event** type, based on the auditable event definitions of the functional components included in the PP~~/ST~~, [*information specified in column three of the IPS Events table*] .

Requirement	Auditable Events	Additional Audit Record Contents
FMT_SMF.1/IPS	Modification of an IPS policy element.	Identifier or name of the modified IPS policy element (e.g. which signature, baseline, or known-good/known-bad list was modified).
IPS_ABD_EXT.1	Inspected traffic matches an anomaly-based IPS policy.	Source and destination IP addresses.
		The content of the header fields that were determined to match the policy.
		TOE interface that received the packet.
		Aspect of the anomaly-based IPS policy rule that triggered the event (e.g. throughput, time of day, frequency, etc.).
		Network-based action by the TOE (e.g. allowed, blocked, sent reset to source IP, sent blocking notification to firewall).
IPS_IPB_EXT.1	Inspected traffic matches a list of known-good or known-bad addresses applied to an IPS policy.	Source and destination IP addresses (and, if applicable, indication of whether the source and/or destination address matched the list).
		TOE interface that received the packet.
		Network-based action by the TOE (e.g. allowed, blocked, sent reset).
IPS_NTA_EXT.1	Modification of which IPS policies are active on a TOE interface. Enabling/disabling a TOE interface with IPS policies applied. Modification of which mode(s) is/are active on a TOE interface.	Identification of the TOE interface.
		The IPS policy and interface mode (if applicable).
IPS_SBD_EXT.1	Inspected traffic matches a signature-based IPS rule with logging enabled.	Name or identifier of the matched signature.
		Source and destination IP addresses.
		The content of the header fields that were determined to match the signature.
		TOE interface that received the packet.
		Network-based action by the TOE (e.g. allowed, blocked, sent reset).

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 Security 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 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.

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-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;
- ECC schemes using "NIST curves" [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;
- FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [selection: RFC 3526, RFC 7919]

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

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 186-4, "Digital Signature Standard (DSS)", Appendix B.3 for RSA schemes;**
- **FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384 and [P-521];**

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

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]**.

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 and cryptographic key sizes (modulus) [2048 bits or greater],*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256 bits or greater]*

] that meet the following: [

- *For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,*

- *For ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves" [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4*].

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]* 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*] hardware-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 using TLS.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [not require client authentication] if the peer certificate is deemed invalid.

FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254 [6668].

FCS_SSHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based [password based].

FCS_SSHS_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_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [aes128-cbc, aes256-cbc].

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [hmac-sha1, hmac-sha2-256, hmac-sha2-512] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [diffie-hellman-group14-sha1] and [no other methods] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

FCS_TLSC_EXT.1 TLS Client protocol

FCS_TLSC_EXT.1.1 The TSF shall implement [TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_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_SHA256 as defined in RFC 5246
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

] 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 SAN, and no other attribute types].

FCS_TLSC_EXT.1.3 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [

- Not implement any administrator override mechanism].

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

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 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_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_SHA256 as defined in RFC 5246
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

] and no other ciphersuites.

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [none].

FCS_TLSS_EXT.1.3 The TSF shall perform key establishment for TLS using [Diffie-Hellman parameters with size [2048 bits], ECDHE curves [secp256r1] and no other curves]].

FCS_TLSS_EXT.1.4 The TSF shall support [session resumption based on session tickets according to RFC 5077].

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 [transport mode, tunnel 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-128 (RFC 3602), AES-CBC-256 (RFC 3602), AES-GCM-128 (RFC 4106), AES-GCM-256 (RFC 4106)] and [no other algorithm] together with a Secure Hash Algorithm (SHA)-based HMAC [HMAC-SHA-256].

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 5996 and [with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)], and [RFC 4868 for hash functions]].

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [IKEv1, IKEv2] protocol uses the cryptographic algorithms [AES-CBC-128, 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 within [*120 seconds to 48] hours*;];
- IKEv2 SA lifetimes can be configured by a Security Administrator based on [
    - length of time, where the time values can be configured within [*120 seconds to 48] hours*]].

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 within [*120 seconds to 48] 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 within [*120 seconds to 48] hours*]].

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 Group(s)

- 19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and [
- [14 (2048-bit MODP)] according to RFC 3526,].

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 all IKE protocols perform peer authentication using [RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [Pre-shared Keys].

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 type].

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:

- Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [*all other printable ASCII characters*]];
- Minimum password length shall be configurable to [*6*] and [*128]*

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 [no other protocols].

FIA_PSK_EXT.1.2 The TSF shall be able to accept text-based pre-shared keys that:

- are 22 characters and [[*between 6 and 128 characters*]];

- composed of any combination of upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”).

FIA_PSK_EXT.1.3 The TSF shall condition the text-based pre-shared keys by using [SHA-1, [*the TOE converts the text string into an authentication value as per RFC 2409 for IKEv1 or RFC 4306 for IKEv2 using the pseudo-random function that is configured as the hash algorithm for the IKE exchanges*]].

FIA_PSK_EXT.1.4 The TSF shall be able to [accept] bit-based pre-shared keys.

FIA_UAU_EXT.2 Password-based authentication mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local [password-based, SSH public key-based] authentication mechanism to perform local administrative user authentication.

FIA_UAU.7 Protected authentication feedback

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_X509_EXT.1/Rev X509 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 TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
- *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
- *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

FIA_X509_EXT.1.2/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 X509 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*].

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 X509 certificate requests

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

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 (trusted update)

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

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 locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using* ***digital signature and*** *[no other] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- ***Ability to manage the cryptographic keys;***
- ***Ability to configure the cryptographic functionality,***
- ***Ability to configure the lifetime for IPsec SAs,***
- ***Ability to import X.509v3 certificates to the TOE's trust store;***

  [
  - *Ability to configure the reference identifier for the peer*
  - Ability to set the time which is used for time-stamps;*].*

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/IPS Specification of management functions (IPS)

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

- *Enable, disable signatures applied to sensor interfaces, and determine the behavior of IPS functionality*
- *Modify these parameters that define the network traffic to be collected and analyzed:*
  - *Source IP addresses (host address and network address)*
  - *Destination IP addresses (host address and network address)*
  - *Source port (TCP and UDP)*
  - *Destination port (TCP and UDP)*

  - *Protocol (IPv4 and IPv6)*
  - *ICMP type and code*
- *Update (import) signatures*
- *Create custom signatures*
- *Configure anomaly detection*
- *Enable and disable actions to be taken when signature or anomaly matches are detected*
- *Modify thresholds that trigger IPS reactions*
- *Modify the duration of traffic blocking actions*
- *Modify the known-good and known-bad lists (of IP addresses or address ranges)*
- *Configure the known-good and known-bad lists to override signature-based IPS policies*]

FMT_SMF.1/VPN Specification of Management Functions (VPN Gateway)

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 locally;*
- *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 demonstrate the correct operation of the TSF: [

- *CPU and Memory BIOS self-tests;*
- *Boot loader image verification;*
- *FIPS 140-2 Known Answer Tests (KAT); and*
- *Noise source tests*].

FPT_TST_EXT.3 TSF 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 updates

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 software/firmware 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 a means to authenticate firmware/software updates to the TOE using a **digital signature mechanism and [no other mechanisms]** prior to installing those updates.

FPT_SKP_EXT.1 Protection of TSF data (for reading of all pre-shared, symmetric and private 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].

FPT_FLS.1/SelfTest Fail secure (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*.]

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 **Administrator**-initiated termination of the **Administrator's** own interactive session.

FTA_TAB.1 Default TOE access banners

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

5.3.8 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 **authorized IT entities supporting the following capabilities: audit server, [no other capabilities]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from **disclosure and detection of modification of the channel data**.

FTP_ITC.1.2 The TSF shall permit **the TSF, or the authorized IT entities** to initiate communication via the trusted channel.

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for *[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/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** 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** 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*
  - *[IPv6 Extension header type [Hop-by-Hop Options, Destination Options, Routing, Fragment, Authentication Header and No Next Header]]*
- *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

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 2460)
  - 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 Traffic Filtering rules: permit and drop with the capability to log the operation.

FPF_RUL_EXT.1.4 The TSF shall allow the Packet Traffic 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.

5.3.11 Intrusion prevention (IPS)

IPS_ABD_EXT.1 Anomaly-based IPS functionality

IPS_ABD_EXT.1.1 The TSF shall support the definition of [baselines ('expected and approved'), anomaly ('unexpected') traffic patterns] including the specification of [

- throughput ([number of bytes or packets per time period (seconds/minutes/hours)]);
- time of day;
- frequency;]

and the following network protocol fields:

- [all packet header and data elements defined in IPS_SBD_EXT.1]

IPS_ABD_EXT.1.2 The TSF shall support the definition of anomaly activity through [manual configuration by administrators].

IPS_ABD_EXT.1.3 The TSF shall allow the following operations to be associated with anomaly-based IPS policies:

- In any mode, for any sensor interface: [
  - allow the traffic flow]
- In inline mode:
  - [allow the traffic flow
  - block/drop the traffic flow
  - and [no other actions]]

IPS_IPB_EXT.1 IP blocking

IPS_IPB_EXT.1.1 The TSF shall support configuration and implementation of known-good and known-bad lists of [source, destination] IP addresses and [no additional address types].

IPS_IPB_EXT.1.2 The TSF shall allow [Security Administrators] to configure the following IPS policy elements: [known-good list rules, known-bad list rules, IP addresses, [*Access Control Lists*]].

IPS_NTA_EXT.1 Network traffic analysis

IPS_NTA_EXT.1.1 The TSF shall perform analysis of IP-based network traffic forwarded to the TOE's sensor interfaces, and detect violations of administratively-defined IPS policies.

IPS_NTA_EXT.1.2 The TSF shall process (be capable of inspecting) the following network traffic protocols:

- [*Internet Protocol (IPv4), RFC 791*
- *Internet Protocol version 6 (IPv6), RFC 2460*
- *Internet control message protocol version 4 (ICMPv4), RFC 792*
- *Internet control message protocol version 6 (ICMPv6), RFC 2463*
- *Transmission Control Protocol (TCP), RFC 793*
- *User Data Protocol (UDP), RFC 768*].

IPS_NTA_EXT.1.3 The TSF shall allow the signatures to be assigned to sensor interfaces configured for promiscuous mode, and to interfaces configured for inline mode, and support designation of one or more interfaces as 'management' for communication between the TOE and external entities without simultaneously being sensor interfaces.

- Promiscuous (listen-only) mode: [*all interfaces on which network traffic can be received and are configured with an IPS policy*];
- Inline (data pass-through) mode: [*all interfaces on which network traffic can be received and are configured with an IPS policy*];
- Management mode: [*all interfaces on which network traffic can be received and are configured with an IPS policy*];
- [no other interface types].

IPS_SBD_EXT.1 Signature-based IPS functionality

IPS_SBD_EXT.1.1 The TSF shall support inspection of packet header contents and be able to inspect at least the following header fields: [

- *IPv4: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; IP Options and [no other field].*
- *IPv6: Version; payload length; next header; hop limit; source address; destination address; routing header; and [traffic class, flow label].*
- *ICMP: Type; Code; Header Checksum; and [ID, sequence number], [Rest of Header (varies based on the ICMP type and code].*
- *ICMPv6: Type; Code; and Header Checksum.*
- *TCP: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options.*
- *UDP: Source port; destination port; length; and UDP checksum*].

IPS_SBD_EXT.1.2 The TSF shall support inspection of packet payload data and be able to inspect at least the following data elements to perform string-based pattern-matching:

- *ICMPv4 data: characters beyond the first 4 bytes of the ICMP header.*
- *ICMPv6 data: characters beyond the first 4 bytes of the ICMP header.*
- *TCP data (characters beyond the 20 byte TCP header), with support for detection of:*

  - *FTP (file transfer) commands: help, noop, stat, syst, user, abort, acct, allo, appe, cdup, cwd, dele, list, mkd, mode, nlst, pass, pasv, port, pass, quit, rein, rest, retr, rmd, rnfr, rnto, site, smnt, stor, stou, stru, and type.*
  - *HTTP (web) commands and content: commands including GET and POST, and administrator-defined strings to match URLs/URIs, and web page content.*
  - *SMTP (email) states: start state, SMTP commands state, mail header state, mail body state, abort state.*
  - *[no other types of TCP payload inspection];*
- *UDP data: characters beyond the first 8 bytes of the UDP header;*
- *[no other types of packet payload inspection]].*

IPS_SBD_EXT.1.3 The TSF shall be able to detect the following header-based signatures (using fields identified in IPS_SBD_EXT.1.1) at IPS sensor interfaces: [

- *IP Attacks*
  - *IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack)*
  - *IP source address equal to the IP destination (Land attack)*
- *ICMP Attacks*
  - *Fragmented ICMP Traffic (e.g. Nuke attack)*
  - *Large ICMP Traffic (Ping of Death attack)*
- *TCP Attacks*
  - *TCP NULL flags*
  - *TCP SYN+FIN flags*
  - *TCP FIN only flags*
  - *TCP SYN+RST flags*
- *UDP Attacks*
  - *UDP Bomb Attack*
  - *UDP Chargen DoS Attack].*

IPS_SBD_EXT.1.4 The TSF shall be able to detect all the following traffic-pattern detection signatures, and to have these signatures applied to IPS sensor interfaces: [

- *Flooding a host (DoS attack)*
  - *ICMP flooding (Smurf attack, and ping flood)*
  - *TCP flooding (e.g. SYN flood)*
- *Flooding a network (DoS attack)*
- *Protocol and port scanning*
  - *IP protocol scanning*
  - *TCP port scanning*
  - *UDP port scanning*
  - *ICMP scanning*].

IPS_SBD_EXT.1.5 The TSF shall allow the following operations to be associated with signature-based IPS policies:

- In any mode, for any sensor interface: [
  - allow the traffic flow;]
- In inline mode:
  - block/drop the traffic flow;
  - and [allow all traffic flow]

IPS_SBD_EXT.1.6 The TSF shall support stream reassembly or equivalent to detect malicious payload even if it is split across multiple non-fragmented packets.

5.4 Assurance Requirements

18 The TOE security assurance requirements are summarized in Table 23.

Table 23: Assurance Requirements

Assurance Class	Components	Description
Security Target Evaluation	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Components Definition
	ASE_INT.1	ST Introduction
	ASE_OBJ.1	Security Objectives for the operational environment
	ASE_REQ.1	Stated Security Requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests	ATE_IND.1	Independent Testing - conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

19 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/IPS, FAU_GEN.2, FAU_STG_EXT.1

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

21 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.

22 Logs are written to the FortiGate unit hard disk if the unit contains one. Models that do not contain a hard disk log to system memory. The amount of audit data that can be stored is dependent on the capacity of the device (see Table 4).

23 If a hypervisor is used to deploy a TOE virtual appliance, then the logs are written to the hypervisor's storage, otherwise they are written to the public cloud's storage.

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, log data is cached prior to transmission. As such, no modification or deletion of the log data is possible. This data is transmitted via TLS.

26 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.

27 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.

b) **Generate CSR.** Action and key reference.

c) **Import Certificate.** Action and key reference.

d) **Import CA Certificate.** Action and key reference.

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

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

Table 24: Key Generation Methods

Method	Key Size (bits)	Curves	Standard
RSA	2048 >	N/A	FIPS 186-4, Appendix B.3 The TOE implements all "shall" and "should" statements and does not implement any '"shall not" " or "should not" statements. Details of "should" statements:

Method	Key Size (bits)	Curves	Standard
			• 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 186-4, Appendix B.4 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 Diffie-Hellman group 14	2048	N/A	RFC 3526, Section 3

Table 25: Key Establishment Methods

Method	Usage	Services
Elliptic-curve schemes	Used in TLS and IPsec. TOE is both sender and receiver.	TLS (Audit Server) TLS/HTTPS (GUI) IPsec (VPN)
Diffie-Hellman group 14	Used in TLS, SSH and IPsec. The TOE meets RFC 3526 Section 3 by implementing the 2048-bit Modular Exponential (MODP) Group.	TLS (Audit Server) TLS/HTTPS (GUI) IPsec (VPN) SSH (Admin CLI)

Table 26: 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	n/a	ISO 18033-3 ISO 10116 ISO 19772
Signature generation and verification	RSA	2048	n/a	n/a	FIPS 186-4 ISO/IEC 9796-2
	ECDSA	256 384 521	n/a	n/a	FIPS 186-4 ISO/IEC 14888-3

Operation	Algorithm	Key size(bits)	Digest size	Block size	Standard(s)
Hashing	SHA	n/a	160 256 384 512	n/a	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

29 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; and

e) Hashing of passwords in non-volatile storage.

6.2.2 Keys and CSPs

30 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.

31 Zeroization of cryptographic keys is performed via the OS kernel and invoked via the Command Line Interface (CLI). In all cases, keys and passwords cannot be viewed through an interface designed specifically for that purpose.

32 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 27: Keys and CSPs

Key/CSP	Storage location and method	Usage	Zeroization
IPsec Manual Authentication Key	AES encrypted in Flash	Used as IPsec Session Authentication Key	Overwritten with zeroes when no longer needed.
IPSec Manual Encryption Key	Plaintext in RAM	Used as IPsec Session Encryption Key using AES (128-, 256-bit)	Overwritten with zeroes when no longer needed.
IPSec Session Authentication Key	Plaintext in RAM	IPsec peer-to-peer authentication using HMAC SHA-1 or 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 Pre-Shared Key	AES encrypted in Flash	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	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 peerto-peer key negotiation using or AES (128-, 256-bit)	Overwritten with zeroes when no longer needed.
IKE RSA Key	Plaintext in Flash (generated with CSR or imported)	Used to generate IKE protocol keys (2048- and 3072-bit signatures)	Overwritten with zeroes when no longer needed.
IKE ECDSA Key	Plaintext in Flash (generated with CSR or imported)	Used to generate IKE protocol keys (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	Overwritten with zeroes when no longer needed.

Key/CSP	Storage location and method	Usage	Zeroization
		new firmware versions using RSA public key	
HTTPS/TLS Server/Host Key	Plaintext in Flash	RSA private key 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- or 3072-bit)	Overwritten with zeroes when no longer needed.
SSH Session Authentication Key	Plaintext in RAM	HMAC SHA-1 or HMAC SHA-256 key used for SSH session authentication	Overwritten with zeroes when no longer needed.
SSH Session Encryption Key	Plaintext in RAM	AES (128-, 256-bit) key used for SSH session encryption	Overwritten with zeroes when no longer needed.
Locally Stored Passwords	AES-128 in configuration file (if FIPS mode is enabled) SHA-1 hash in Flash (if FIPS mode is not enabled)	User authentication	Overwritten with zeroes when no longer needed.
Configuration Encryption Key	Plaintext in Flash	AES 256-bit key used to encrypt CSPs on the Boot device	Overwritten with zeroes when no longer needed.

6.2.3 Entropy and DRBG

33 As shown in Table 4 (Entropy column), the entropy source in use varies between TOE models and can be one of:

a) **Token.** Wide-band radio frequency (RF) white noise source provided by the Fortinet Entropy Token. In the case of virtual appliances, USB passthrough provides access to the token entropy source.

b) **SoC3.** Oscillator based entropy source.

c) **CP9.** Oscillator based entropy source.

34 Additional detail regarding these entropy sources is provided the proprietary Entropy Description.

35 In all models, the TOE contains a CTR_DRBG that is seeded from a hardware entropy source. Entropy from the noise source is extracted, conditioned and used to seed the DRBG with 256 bits of full entropy.

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

36 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.

37 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 setup 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

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

39 The server only allows TLS protocol versions 1.1 and 1.2 (rejecting any other protocol version, including SSL 2.0, SSL 3.0 and TLS 1.0 and any other unknown TLS version string supplied) and is restricted to the following ciphersuites:

a) TLS_DHE_RSA_WITH_AES_128_CBC_SHA

b) TLS_DHE_RSA_WITH_AES_256_CBC_SHA

c) TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

d) TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

e) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

f) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

g) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA

h) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA

i) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

j) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

k) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

l) TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

40 Ciphersuites are not user-configurable.

41 The TLS server is capable of negotiating ciphersuites that include DHE and ECDHE key agreement schemes. The DHE key agreement parameters are restricted to 2048 bits and are hardcoded into the server. The ECDHE key agreement parameters use secp256r1 and are hardcoded into the server.

42 The TLS server supports session tickets. Session tickets adhere to the structural format provided in section 4 of RFC 5077. Session tickets are encrypted according to the TLS negotiated symmetric encryption algorithm.

43 Session resumption and establishment require session IDs only.

6.3.3 TLS Client

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

45 TLS 1.1 and 1.2 are allowed and ciphersuites are restricted to:

a) TLS_DHE_RSA_WITH_AES_128_CBC_SHA

b) TLS_DHE_RSA_WITH_AES_256_CBC_SHA

c) TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

d) TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

e) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

f) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

g) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA

h) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA

i) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

j) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

k) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

l) TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

46 Ciphersuites are not user-configurable.

47 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).

48 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.

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

50 The TLS client will transmit the Supported Elliptic Curves extension in the Client Hello message by default with support for the following NIST curves: P256. The non-TOE server can choose to negotiate the elliptic curve from this set for any of the mutually negotiable elliptic curve ciphersuites.

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

6.4 SSH

SFRs:	FCS_SSHS_EXT.1

52 The TOE implements SSH in compliance with RFCs 4251 through 4254 and 6668.

53 The TOE supports password-based or public key (SSH-RSA) authentication.

54 The TOE examines the size of each received SSH packet. If the packet is greater than 256KB, it is automatically dropped.

55 The TOE utilizes AES-CBC-128 and AES-CBC-256 for SSH encryption.

56 The TOE provides data integrity for SSH connections via HMAC-SHA1, HMAC-SHA2-256 and HMAC-SHA2-512.

57 The TOE supports Diffie-Hellman Group 14 SHA-1 (diffie-hellman-group14-sha1) for SSH key exchanges.

58 The TOE will re-key SSH connections after 1 hour or after an aggregate of 1 gig of data has been exchanged (whichever occurs first).

59 The TOE establishes a user identity by verifying that the SSH client's present public key matches the one that is stored within the SSH server's authorized keys file.

6.5 IPsec

SFRs:	FCS_IPSEC_EXT.1

60 The TOE implements IPsec in accordance with RFC 4301.

61 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 send 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.

62 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).

63 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.

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

65 The TOE implements the ESP protocol as defined in RFC 4301. The TOE implements AES-CBC-128 and AES-CBC-256 (per RFC 3602) and AES-GCM-128 and AES-GCM-256 (per RFC 4106) in conjunction with a Secure Hash Algorithm-based HMAC to provide encryption services for ESP.

66 The TOE implements both IKEv1 (as defined in RFCs 2407, 2408, 2409 and 4109 with RFC 4304 for extended sequence numbers) and IKEv2 (as defined in RFC 5996, with mandatory support for NAT traversal as specified in RFC 5996 and RFC 4868 for hash functions).

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

68 The TOE implements AES-CBC-128 and AES-CBC-256 (per RFC 3602) to provide payload encryption for IKEv1 and IKEv2.

69 The TOE permits the configuration of IKEv1 Phase 1 SA and IKEv2 SA lifetimes in seconds, between 120 and 172800 (48 hours).

70 The TOE permits the configuration of IKEv1 Phase 2 SA and IKEv2 Child SA lifetimes in number of bytes or seconds, between 120 and 172800 (48 hours).

71 The TOE utilises 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-1 and SHA-256;

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

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

73 The TOE provides encryption algorithms with a strength between 128 and 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 SA (IKEv2) connection.

74 The TOE permits peer authentication via RSA or ECDSA public keys (X509v3 certificates that conform to RFC 4945) or pre-shared keys.

75 The TOE accepts text-based pre-shared keys that are between 6 and 128 characters in length and composed of any combination of upper and lower case letters, numbers, and special characters (as specified in FIA_PSK_EXT.1.2).

76 The TOE accepts bit-based pre-shared keys.

77 The TOE converts text-based pre-shared keys into an authentication value as per RFC 2409 for IKEv1 or RFC 4306 for IKEv2, using SHA-1 or the PRF that is configured as the hash algorithm for the IKE exchanges.

78 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 DN reference identifiers.

6.6 Residual Data Protection

SFRs:	FDP_RIP.2

79 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.

6.7 Identification and Authentication

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

80 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.

81 The TOE enforces a password policy. Administrative passwords may be at least 8 characters and at most 64 characters long and may be comprised of any combination of upper and lower case letters, numbers, and the following special characters: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)” and all other printable ASCII characters.

82 Administrators connecting via a local connection (console) or remote (HTTPS/TLS or SSH) must provide a valid username and password to complete authentication. 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 (TLS/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 ssh-rsa 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 ssh-rsa authentication succeeds), the TOE shall end the logon process and give the administrator access to TOE functionality (a successful logon).

6.8 X509 Certificates

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

83 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).

84 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;

85 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.

86 Certificate revocation checking for the above scenarios is performed using a CRL.

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

88 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.

89 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.

90 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

91 If, during the entire trust chain verification activity, any certificate under review fails a verification check, then the entire trust chain is deemed untrusted.

92 As part of the verification process, CRL is used to determine whether the certificate is revoked or not. If the CRL cannot be obtained, then the TOE will use the last cached information available about certificate to accept or reject the certificate. 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.

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

94 For the Certificate Signing Request, a CN is required and may be an IP address, DNS name or email address. SANs are optional and may be email, IP address, URI, DNS name or directory name.

6.9 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/IPS, FMT_SMF.1/FFW, FMT_SMF.1/VPN, FMT_SMR.2

95 The TOE does not permit access to any functions (other than the warning/consent banner and authentication interface) prior to login. The TOE version is displayed at the GUI prior to authentication.

96 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 and the CLI. The Security Administrator is able to perform the following functions:

a) Administer the TOE locally and remotely;

b) Configure the access banner;

c) Configure the session inactivity time before session termination or locking;

d) Update the TOE, and to verify the updates using digital signature capability prior to installing those updates;

e) Configure the cryptographic functionality;

f) Generate and delete cryptographic keys. In particular, a security administrator can generate and delete the cryptographic keys associated with CSRs;

g) Configure the IPsec functionality;

h) Import X.509v3 certificates;

i) Enable, disable, determine and modify the behavior of all the security functions of the TOE identified in this EP to the Administrator;

j) Ability to configure firewall rules;

k) Enable, disable signatures applied to sensor interfaces, and determine the behavior of IPS functionality;

l) Ability to modify (enable/disable) transmission of audit records to an external audit server;

m) Ability to set the time;

n) Modify these parameters that define the network traffic to be collected and analyzed:

  i) Source IP addresses (host address and network address)

  ii) Destination IP addresses (host address and network address)

  iii) Source port (TCP and UDP)

  iv) Destination port (TCP and UDP)

  v) Protocol (IPv4 and IPv6)

  vi) ICMP type and code

o) Update (import) signatures;

p) Create custom signatures;

q) Configure anomaly detection;

r) Enable and disable actions to be taken when signature or anomaly matches are detected;

s) Modify thresholds that trigger IPS reactions;

t) Modify the duration of traffic blocking actions;

u) Modify the known-good and known-bad lists (of IP addresses or address ranges); and

v) Configure the known-good and known-bad lists to override signature-based IPS policies.

w) Ability to restart SSHD, HTTPSD, OFTPD services.

6.10 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

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

98 Pre-shared keys related to 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 with the encrypted password hashes 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.

99 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 Known Answer Tests (KAT)

i) Comparison of a number of cryptographic functions against an expected set of values

100 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.

101 The cryptographic functionality will not be available if the cryptographic tests fail, and any operation of the TOE supported by this functionality will not be available. If the CPU, or BIOS tests fail, the device will not complete the boot up operation. If the boot loader image verification fails, the boot up operation will fail. When the device completes the boot up operation, this is evidence that the self-tests have passed, and that the TOE, and the cryptographic functions are operating correctly.

102 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.

103 The administrator may query the current version of the TOE via the GUI or CLI. The TOE will notify administrators if a new update file is available, but the update process will not commence until requested by the administrator.

104 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.

105 The TOE maintains its own time source, which is free from outside interference. The Security Administrator sets the date and time during initial TOE configuration and may change the time during operation. This timestamp is used for the purposes of generating audit logs and other time-sensitive operations on the TOE including cryptographic key regeneration intervals.

6.10.1 TOE Initialization

106 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

c) Loading and Initialization of:

i) Kernel;

ii) Firmware;

iii) Cryptographic known answer tests;

iv) Entropy gathering and DRBG initialization; and

v) Cryptographic module.

107 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.

6.11 TOE Access

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

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

109 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.

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

6.12 Trusted Path/Channels

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

111 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).

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

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

a) TLS (Versions 1.1 and 1.2) and HTTPS (in compliance with RFC 2818) for the Web GUI; and

b) SSH in compliance with the following RFCs: 4251, 4252, 4253, 4254 and 6668.

114 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.

6.13 Stateful Traffic/Packet Filtering

SFRs:	FFW_RUL_EXT.1, FPF_RUL_EXT.1

115 The TOE permits the configuration of stateful packet filtering policies. The TOE supports the full range of Transport Layer Protocols for both IPv4 and IPv6. 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 2460)
   i) Source address;
   ii) Destination Address;
   iii) Transport Layer Protocol; and
   iv) The following IPv6 Extension header types:
      - Hop-by-Hop Options;
      - Destination Options;
      - Routing;
      - Fragment;
      - Authentication Header; and
      - No Next Header.

e) TCP (RFC 793)
   i) Source Port; and
   ii) Destination Port

f) UDP (RFC 768)
   i) Source Port; and
   ii) Destination Port

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

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

118 Each packet that arrives on an interface is subject to the enforcement of the 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.

119 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.

120 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.

121 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.

122 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.

123 The TOE utilizes a session database to track active sessions for TCP, UDP and ICMP (amongst other protocols). A number of variables (such as source/destination address and ports, sequence numbers, flags and TTL values) are utilized in the management of sessions.

124 Periodically old sessions exceeding their TTL are removed from the database. Sessions that have been closed are similarly removed from the database.

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

126 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).

127 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.

128 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.

129 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.

130 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).

131 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, VPN processes and IPS processes.

132 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.

133 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.

6.14 Intrusion Prevention (IPS)

SFRs:	IPS_ABD_EXT.1, IPS_IPB_EXT.1, IPS_NTA_EXT.1, IPS_SBD_EXT.1

134 As part of the evaluated configuration, IPS packet/event logging should be enabled by administrators for each rule/signature they enable.

135 The IPS engine will automatically group similar IPS events based on timer periods. Repeat events for a single session will be grouped and logged every 5 seconds. Upon termination of a session, all unreported events (after duplication elimination) will be logged.

136 Security policies (both IPv4 and IPv6) can be assigned to any TOE interface capable of receiving network traffic and a single policy can be assigned to more than one interface, both physical and virtual. As such, when an administrator includes an IPS policy within a security policy, that IPS policy is applied to whichever network interfaces are included in the base policy.

137 Good/bad address lists are constructed as part of IPv4/IPv6 policies. Single addresses, address groups or whole subnets can be attached to a policy, along with policy-related attributes (assigned IPS policies, permit/deny/IPsec reactions, etc.), which will subsequently dictate how the TOE reacts to traffic matching the policy.

138 Access Control Lists (ACLs) may also be defined, and can be considered a granular or more specifically targeted blacklist. ACLs drop IPv4 or IPv6 packets at the physical network interface before the packets are analyzed by the CPU. The only action available for an ACL policy is ‘deny’. ACLs are defined based on:

a) the incoming interface;

b) the source addresses of the traffic;

c) the destination addresses of the traffic;

d) the services or ports the traffic is using (services are pre-defined well known protocols).

139 Access to IPS functionality is provided to all users assigned to an Administrator profile. The administrator profile has full privileges for configuring and activating IPS policies.

140 Active security policies are enforced in a sequential order (based on the sequence number assigned to each policy). The administrator may change this sequence to suit their needs. If an IPS policy is attached to a security policy, it will be enforced at the same time as the security policy.

141 The default Implicit Deny policy will be enforced if the received traffic does not match any of the other active policies.

142 IPS rules can be associated with the following protocols:

a) Internet Protocol (IPv4), RFC 791;

b) Internet Protocol version 6 (IPv6), RFC 2460;

c) Internet control message protocol version 4 (ICMPv4), RFC 792;

d) Internet control message protocol version 6 (ICMPv6), RFC 2463;

e) Transmission Control Protocol (TCP), RFC 793; and

f) User Data Protocol (UDP), RFC 768.

143 The TOE’s conformance with the protocols/RFCs listed above is determined during development. Compliance testing is performed as part of the development and release process with changes being made as required to ensure conformance.

144 IPS policies can be applied to any TOE interface that can be included in an IPv4 or IPv6 policy. All interfaces capable of receiving network traffic can be utilised as a sensor (promiscuous or inline), management or other type of interface as defined in the IPS EP.

145 A ‘management’ interface, in the context of the IPS EP, can be considered any TOE interface configured for remote administration that does not have an IPS policy tied to it.

146 A signature rule is comprised of administrator-selected components which specify the parameters (traffic class, protocol, ports, etc.) that must be met to trigger a reaction (allow, drop, reset, etc.). Signature rules are authored in the following format:

a) `F-SBID(--parameter1; --parameter2; --etc);`

147 An administrator may include as many fields/variables as required to fully define the signature.

148 The TOE supports the configuration of baseline and anomaly-based attributes when constructing signatures for inclusion in IPS policies. The following parameters may be included as part of the definition of baselines/anomalous traffic patterns in the format indicated above:

a) Throughput (number of bytes or packets per time period (seconds/minutes/hours);

b) Time of day; and

c) Frequency.

149 In addition to the above, the TOE permits the use of the following header fields and data payloads for each supported protocol:

a) **IPv4**: Version; Header Length; Packet Length; ID; IP Flags; Fragment Offset; Time to Live (TTL); Protocol; Header Checksum; Source Address; Destination Address; and IP Options.

b) **IPv6**: Version; traffic class; flow label; payload length; next header; hop limit; source address; destination address; routing header; home address options.

c) **ICMP**: Type; Code; Header Checksum; and Rest of Header (varies based on the ICMP type and code.

   i) ICMPv4 data: characters beyond the first 4 bytes of the ICMP header.

d) **ICMPv6**: Type; Code; and Header Checksum.

   i) ICMPv6 data: characters beyond the first 4 bytes of the ICMP header.

e) **TCP**: Source port; destination port; sequence number; acknowledgement number; offset; reserved; TCP flags; window; checksum; urgent pointer; and TCP options.

   i) TCP data (characters beyond the 20 byte TCP header), with support for detection of:

f) **FTP** (file transfer) commands: help, noop, stat, syst, user, abort, acct, allo, appe, cdup, cwd, dele, list, mkd, mode, nlst, pass, pasv, port, pass, quit, rein, rest, retr, rmd, rnfr, rnto, site, smnt, stor, stou, stru, and type.

g) **HTTP** (web) commands and content: commands including GET and POST, and administrator-defined strings to match URLs/URIs, and web page content; and

h) **SMTP** (email) states: start state, SMTP commands state, mail header state, mail body state, abort state.

i) **UDP**: Source port; destination port; length; and UDP checksum.

   i) UDP data: characters beyond the first 8 bytes of the UDP header.

150 The TOE supports stream reassembly and is capable of detecting malicious payloads even if they are split across multiple fragmented packets.

151 Each IPS rule can be configured with a reaction the TOE should take when a match is identified once added to an IPS Policy – allowing the traffic flow, denying the traffic flow or sending a TCP reset to the traffic source.

152 On some hardware models ports are tagged with names such as “WiFi”, “LAN”, “WAN” or “DMZ” either through identification on the user interface or through screened graphics on the TOE hardware. Tagging of the interfaces in this manner does not affect the TOE’s capability for the

enforcement of SFRs as all rules are capable of being configured on all types of interface. Identifying interfaces in this manner is done for the purposes of simplifying administration and identifying the correct port as well as applying a suitable default configuration. Details as to the default configuration for each interface type as well as the methods for modifying their configuration can be found in the administrative guidance.

153 String-based detection signatures follow the format defined previously but will include a '—pattern' field that allows administrators to define regular expressions for string matching.

154 Packet payload (string-based) detection signatures must be placed within an IPS policy before being configured with the reactions specified earlier (allow, deny or reset).

155 The TOE is capable of detecting the following attack and scan types:

a) IP Attacks
   i) IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack); and
   ii) IP source address equal to the IP destination (Land attack).

b) ICMP Attacks
   i) Fragmented ICMP Traffic (e.g. Nuke attack); and
   ii) Large ICMP Traffic (Ping of Death attack).

c) TCP Attacks
   i) TCP NULL flags;
   ii) TCP SYN+FIN flags;
   iii) TCP FIN only flags; and
   iv) TCP SYN+RST flags.

d) UDP Attacks
   i) UDP Bomb Attack; and
   ii) UDP Chargen DoS Attack.

e) Flooding a host (DoS attack)
   i) ICMP flooding (Smurf attack, and ping flood);
   ii) TCP flooding (e.g. SYN flood); and
   iii) Flooding a network (DoS attack).

f) Protocol and port scanning
   i) IP protocol scanning
   ii) TCP port scanning;
   iii) UDP port scanning; and
   iv) ICMP scanning.

156 These attacks are processed in the same manner as all other traffic passing through the TOE. The reaction is configurable by the administrator and can be set to Permit, Deny or Reset (for TCP).

7 Rationale

7.1 Conformance Claim Rationale

157 The following rationale is presented with regard to the PP/EP conformance claims:

a) **TOE type.** As identified in section 2.1, the TOE is firewall with IPS and VPN capabilities consistent with the claimed PP/EPs.

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

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

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

7.2 Security Objectives Rationale

158 All security objectives are drawn directly from the claimed PP/EPs.

7.3 Security Requirements Rationale

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

Annex A: Extended Components Definition

160 See appended Protection Profile extracts.

161 The following extended components are taken from the IPS_EP which does not provide an extended components definition:

a) IPS_ABD_EXT.1

b) IPS_IPB_EXT.1

c) IPS_NTA_EXT.1

d) IPS_SBD_EXT.1

Annex B: CAVP Certificates

Annex B.1: SFR Coverage

Table 28: CAVP SFR Coverage Mapping

SFR	Selections	Usage	CAVP	Notes
FCS_CKM.1 Cryptographic Key Generation	RSA	TLS, SSH	C1576 C2199 C2201	
			C1578 C1797 C1798 A1253	Hardware acceleration per Table 29.
	ECC (ECDSA)	TLS	C1576 C2199 C2201	
			C1578 C1798 A1253	Hardware acceleration per Table 29.
	FFC – DH Group 14	IPsec, TLS, SSH	n/a	No NIST CAVP, CCTL must perform all assurance/evaluation activities.
FCS_CKM.1/IKE Cryptographic Key Generation (for IKE Peer Authentication)	RSA	IPsec	C1576 C2199 C2201	Certificate / key gen performed by SSL library.

SFR	Selections	Usage	CAVP	Notes
			C1578 C1797 C1798 A1253	Hardware acceleration per Table 29.
	ECDSA	IPsec	C1576 C2199 C2201	Certificate / key gen performed by SSL library.
			C1578 C1798 A1253	Hardware acceleration per Table 29.
FCS_CKM.2 Cryptographic Key Establishment	ECC	TLS	C1576 C2199 C2201	
		IPsec	C1575 C2197 C2200	
	DH Group 14	IPsec, TLS, SSH	n/a	
FCS_COP.1/DataEncryption	AES-CBC-128/256	TLS, SSH	C1549 C2140	
			C1797 C1798 C1578	Hardware acceleration per Table 29.

SFR	Selections	Usage	CAVP	Notes
			A1253	
		IPsec	C1575 C2197 C2200	
			C1797 C1798 C1578 A1253	Hardware acceleration per Table 29.
	AES-GCM-128/256	TLS, SSH	C1576 C2199 C2201	
			C1798 C1578 A1253	Hardware acceleration per Table 29.
		IPsec	C1575 C2197	
			C1798 C1578 A1253	Hardware acceleration per Table 29.
FCS_COP.1/SigGen	RSA	IPsec, TLS, SSH, Trusted Update	C1576 C2199 C2201	

SFR	Selections	Usage	CAVP	Notes
			C1797 C1798 C1578 A1253	Hardware acceleration per Table 29.
	ECDSA	TLS	C1549	
			C1798 C1578 A1253	Hardware acceleration per Table 29.
		IPsec	C1575 C2197 C2200 C2201	
			C1798 C1578 A1253	Hardware acceleration per Table 29.
FCS_COP.1/Hash	SHA-1, SHA-256, SHA-384, SHA-512	IPsec, Password Hashing	C1575 C2197 C2200	
		TLS, SSH	C1576 C2199 C2200 C2201	

SFR	Selections	Usage	CAVP	Notes
		IPsec, TLS, SSH	C1797 C1798 C1578 A1253	Hardware acceleration per Table 29.
FCS_COP.1/KeyedHash	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	IPsec	C1575 C2197 C2200	
		TLS, SSH	C1576 C2199 C2201	
		IPsec, TLS, SSH	C1798 C1578 A1253	Hardware acceleration per Table 29.
	HMAC-SHA-1, HMAC-SHA-256	IPsec, TLS, SSH	C1797	Hardware acceleration per Table 29.
FCS_RBG_EXT.1	CTR_DRBG (AES)	TOE RBG	C1573 C2195 C2198	

Annex B.2: CAVP Libraries

Table 29: CAVP Libraries & Capabilities Mapping

Library Name	CAVP	Capability	Usage
Fortinet FortiOS FIPS Cryptographic Library v6.2 Fortinet FortiOS-VM FIPS Cryptographic Library v6.2	C1575 C2197 C2200	AES-CBC	Primarily IPsec
		AES-GCM	
		AES-GMAC	
		ECDSA KeyGen (186-4) ECDSA SigGen (186-4) ECDSA SigVer (186-4)	
		HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	
		KAS-ECC	
		KAS-FFC	
		KDF IKEv1	
		KDF IKEv2	
		SHA-1, SHA-256, SHA-384, SHA-512	
Fortinet FortiOS SSL Cryptographic Library v6.2	C1549 C2140	AES-CBC	Primarily TLS and SSH
	C1576	AES-GCM	

Library Name	CAVP	Capability	Usage
Fortinet FortiOS-VM SSL Cryptographic Library v6.2	C2199 C2201	Component SSH	
		Component TLS	
		ECDSA KeyGen (186-4)	
		ECDSA SigGen (186-4) ECDSA SigVer (186-4)	
		HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	
		KAS-ECC Component	
		KAS-FFC Component	
		RSA KeyGen (186-4) RSA SigGen (186-4) RSA SigVer (186-4)	
		SHA-1, SHA-256, SHA-384, SHA-512	
Fortinet FortiOS RBG Cryptographic Library v6.2 Fortinet FortiOS-VM RBG Cryptographic Library v6.2	C1573 C2195 C2198	Counter DRBG	TOE DRBG
	C1797	AES-CBC	

Library Name	CAVP	Capability	Usage
Fortinet CP8 Cryptographic Library		HMAC-SHA-1, HMAC-SHA2-256	Hardware acceleration when available and configured (enabled by default). See CP8 in ASIC column of Table 30.
		RSA KeyGen (186-4) RSA SigGen (186-2) RSA SigGen (186-4) RSA SigVer (186-4)	
		SHA-1, SHA-256, SHA-384, SHA-512	
Fortinet FortiOS CP9 Cryptographic Library	C1578	AES-CBC	Hardware acceleration when available and configured (enabled by default). See CP9 in ASIC column of Table 30.
		AES-GCM	
		AES-GMAC	
		ECDSA KeyGen (186-4) ECDSA SigGen (186-4) ECDSA SigVer (186-4)	
		HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	
		KDF IKEv1	
		KDF IKEv2	
		KDF TLS	
		RSA KeyGen (186-4) RSA SigGen (186-4)	

Library Name	CAVP	Capability	Usage
		RSA SigVer (186-4)	
		SHA-1, SHA-256, SHA-384, SHA-512	
Fortinet CP9Lite Cryptographic Library	C1798	AES-CBC	Hardware acceleration when available and configured (enabled by default). See CP9 lite in ASIC column of Table 30.
		AES-GCM	
		AES-GMAC	
		ECDSA KeyGen (186-4) ECDSA SigGen (186-4) ECDSA SigVer (186-4)	
		HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	
		RSA SigGen (186-4) RSA SigVer (186-4)	
		SHA-1, SHA-256, SHA-384, SHA-512	
Fortinet CP9XLite Cryptographic Library	A1253	AES-CBC	Hardware acceleration when available and configured (enabled by default). See CP9XLite in ASIC column of Table 30.
		AES-GCM	
		ECDSA KeyGen (186-4) ECDSA SigGen (186-4) ECDSA SigVer (186-4)	

Library Name	CAVP	Capability	Usage
		HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	
		KDF IKEv1	
		KDF IKEv2	
		KDF TLS	
		RSA KeyGen (186-4) RSA SigGen (186-4) RSA SigVer (186-4)	
		SHA-1, SHA-256, SHA-384, SHA-512	

Annex B.3: CAVP Hardware Mapping

Table 30: CAVP Hardware Coverage

Model	CPU	Architecture	ASIC	CAVP
FG-30E	Fortinet SoC3	ARMv7-A	n/a	C1575 C1549 C1576 C1573
FGR-30D	Fortinet SoC3	ARMv7-A	n/a	
FWF-30E	Fortinet SoC3	ARMv7-A	n/a	
FG-50E	Fortinet SoC3	ARMv7-A	n/a	
FWF-50E	Fortinet SoC3	ARMv7-A	n/a	
FG-51E	Fortinet SoC3	ARMv7-A	n/a	
FWF-51E	Fortinet SoC3	ARMv7-A	n/a	

Model	CPU	Architecture	ASIC	CAVP
FG-52E	Fortinet SoC3	ARMv7-A	n/a	
FWF-50E-2R	Fortinet SoC3	ARMv7-A	n/a	
FG-1500D-DC	Intel Xeon E5-1650	Ivy Bridge	n/a	C1575 C1549 C1576 C1573
FG-1500DT	Intel Xeon E5-1650v2	Ivy Bridge	n/a	
FG-60E	Fortinet SoC3	ARMv7-A	CP9Lite	C1575 C1549 C1576 C1573 C1798
FG-60E-DSL	Fortinet SoC3	ARMv7-A	CP9Lite	
FG-60E-PoE	Fortinet SoC3	ARMv7-A	CP9Lite	
FG-60F	Fortinet SoC4	ARMv8	CP9Lite	
FWF-60E	Fortinet SoC3	ARMv7-A	CP9Lite	
FWF-60E-DSL	Fortinet SoC3	ARMv7-A	CP9Lite	
FWF-60F	Fortinet SoC4	ARMv8	CP9Lite	
FG-61E	Fortinet SoC3	ARMv7-A	CP9Lite	
FG-61F	Fortinet SoC4	ARMv8	CP9Lite	
FWF-61E	Fortinet SoC3	ARMv7-A	CP9Lite	
FWF-61F	Fortinet SoC4	ARMv8	CP9Lite	
FG-80E	Fortinet SoC3	ARMv7-A	CP9Lite	
FG-80E-PoE	Fortinet SoC3	ARMv7-A	CP9Lite	
FG-81E	Fortinet SoC3	ARMv7-A	CP9Lite	

Model	CPU	Architecture	ASIC	CAVP
FG-81E-PoE	Fortinet SoC3	ARMv7-A	CP9Lite	
FG-100E	Fortinet SoC3	ARMv7-A	CP9Lite	
FG-100EF	Fortinet SoC3	ARMv7-A	CP9Lite	
FG-100F	Fortinet SoC4	ARMv8	CP9Lite	
FG-101E	Fortinet SoC3	ARMv7-A	CP9Lite	
FG-101F	Fortinet SoC4	ARMv8	CP9Lite	
FG-140E	Fortinet SoC3	ARMv7-A	CP9Lite	
FG-60E-DSLJ	Fortinet SoC3	ARMv7-A	CP9Lite	
FG-140E-PoE	Fortinet SoC3	ARMv7-A	CP9Lite	
FWF-60E-DSLJ	Fortinet SoC3	ARMv7-A	CP9Lite	
FG-200E	Intel Celeron G1820	Haswell	CP9	C1575 C1549 C1576 C1573 C1578
FG-201E	Intel Celeron G1820	Haswell	CP9	
FG-300D	Intel i3-3220	Ivy Bridge	CP8	C1575 C1549 C1576 C1573 C1797
FG-300E	Intel i5-6500	Skylake	CP9	C1575

Model	CPU	Architecture	ASIC	CAVP
FG-301E	Intel i5-6500	Skylake	CP9	C1549 C1576 C1573 C1578
FG-400D	Intel i3-3220	Ivy Bridge	CP8	C1575 C1549 C1576 C1573 C1797
FG-400E	Intel i5- 8500	Coffeelake	CP9	C1575 C1549 C1576 C1573 C1578
FG-401E	Intel i5- 8500	Coffeelake	CP9	
FG-500D	Intel Xeon E3-1225v2	Ivy Bridge	CP8	C1575 C1549 C1576 C1573 C1797
FG-500E	Intel i7-6700	Skylake	CP9	C1575 C1549 C1576 C1573 C1578
FG-501E	Intel i7-6700	Skylake	CP9	

Model	CPU	Architecture	ASIC	CAVP
FG-600D	Intel i7-3770	Ivy Bridge	CP8	C1575 C1549 C1576 C1573 C1797
FG-600E	Intel i7-8700	Coffeelake	CP9	C1575 C1549 C1576 C1573 C1578
FG-601E	Intel i7-8700	Coffeelake	CP9	
FG-900D	Intel Xeon E3-1225v3	Haswell	CP8	C1575 C1549 C1576 C1573 C1797
FG-1000D	Intel Xeon E5-1275v3	Haswell	CP8	
FG-1100E	Intel Xeon E-2186G	Coffeelake	CP9	C1575 C1549 C1576 C1573 C1578
FG-1101E	Intel Xeon E-2186G	Coffeelake	CP9	
FG-1200D	Intel Xeon E5-1275v3	Haswell	CP8	C1575 C1549 C1576 C1573
FG-1500D	Intel Xeon E5-1650v2	Ivy Bridge	CP8	

Model	CPU	Architecture	ASIC	CAVP
				C1797
FG-2000E	Intel Xeon E5-1660v4	Broadwell	CP9	C1575 C1549 C1576 C1573 C1578
FG-2200E	Intel Xeon Gold 6126	Purely	CP9	
FG-2201E	Intel Xeon Gold 6126	Purley	CP9	
FG-2500E	Intel Xeon E5-1660v4	Broadwell	CP9	
FG-3000D	Intel Xeon E5-2650v3	Haswell	CP8	C1575 C1549 C1576 C1573 C1797
FG-3100D-DC	Intel Xeon E5-2660 v3	Haswell	CP8	
FG-3200D-DC	Intel Xeon E5-2670 v3	Haswell	CP8	
FG-3100D	Intel Xeon E5-2660v3	Haswell	CP8	
FG-3200D	Intel Xeon E5-2670v3	Haswell	CP8	
FG-3300E	Intel Xeon Gold 5118	Purely	CP9	C1575 C1549 C1576 C1573 C1578
FG-3301E	Intel Xeon Gold 5118	Purley	CP9	
FG-3400E	Intel Xeon Gold 6130	Purely	CP9	
FG-3401E	Intel Xeon Gold 6130	Purley	CP9	
FG-3600E	Intel Xeon Gold 6152	Purley	CP9	
FG-3601E	Intel Xeon Gold 6152	Purely	CP9	
FG-3700D	Intel Xeon E5-2680v2	Ivy Bridge	CP8	C1575 C1549 C1576
FG-3800D	Intel Xeon E5-2680v2	Ivy Bridge	CP8	
FG-3810D	Intel Xeon E5-2680v2	Ivy Bridge	CP8	

Model	CPU	Architecture	ASIC	CAVP
FG-3815D	Intel Xeon E5-2680v2	Ivy Bridge	CP8	C1573 C1797
FG-3960E	Intel Xeon E5-2650v4	Broadwell	CP9	C1575 C1549 C1576 C1573 C1578
FG-3980E	Intel Xeon E5-2680v4	Broadwell	CP9	
FG-5001D*	Intel Xeon E5-2658v2	Ivy Bridge	CP8	C1575 C1549 C1576 C1573 C1797
FG-5001E*	Intel Xeon E5-2690v4	Broadwell	CP9	C1575 C1549 C1576 C1573 C1578
FG-5001E1	Intel Xeon E5-2690v4	Broadwell	CP9	
FGR-60F	Fortinet SoC4	ARMv8	CP9XLite	C1575 C1549 C1576 C1573 A1253
FGR-60F-3G4G	Fortinet SoC4	ARMv8	CP9XLite	
FG-40F	Fortinet SoC4	ARMv8	CP9XLite	C1575

Model	CPU	Architecture	ASIC	CAVP
FG-40F-3G4G	Fortinet SoC4	ARMv8	CP9XLite	C1549 C1576 C1573 A1253
FG-60F	Fortinet SoC4	ARMv8	CP9XLite	
FG-61F	Fortinet SoC4	ARMv8	CP9XLite	
FWF-40F	Fortinet SoC4	ARMv8	CP9XLite	C1575 C1549 C1576 C1573 A1253
FWF-40F-3G4G	Fortinet SoC4	ARMv8	CP9XLite	
FWF-60F	Fortinet SoC4	ARMv8	CP9XLite	
FWF-61F	Fortinet SoC4	ARMv8	CP9XLite	

Annex B.3: CAVP Virtual Appliance Coverage

Table 31: TOE Virtual Appliance and related Hardware

Model	VM	Tested Hypervisor	Tested CPUs	CAVP
FortiGate-VM01	Fortinet FortiOS-VM v6.2	VMware ESXi 6.7	Intel Xeon D-15xx Intel Xeon E3 15xx Intel Xeon E-22xx Intel Core i7-68xx	C2140 C2195 C2197 C2198 C2199 C2200 C2201
FortiGate-VM02				
FortiGate-VM04				
FortiGate-VM08				
FortiGate-VM16				
FortiGate-VM32				
FortiGate-VMUL				

C. Extended Component Definitions

This appendix contains the definitions for the extended requirements that are used in the cPP, including those used in Appendices A and B.

(Note: formatting conventions for selections and assignments in this Appendix are those in [CC2].)

C.1 Security Audit (FAU)

C.1.1 Security Audit Data Generation (FAU_GEN_EXT)

Family Behaviour

This component defines the requirements for components in a distributed TOE to generate security audit data.

Component levelling

FAU_GEN_EXT.1 Security audit data shall be generated by all components in a distributed TOE

Management: FAU_GEN_EXT.1

The following actions could be considered for the management functions in FMT:

a) The TSF shall have the ability to configure the cryptographic functionality.

Audit: FAU_GEN_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) No audit necessary.

C.1.1.1 FAU_ GEN_EXT.1 Security Audit Data Generation for Distributed TOE Components

FAU_GEN_EXT.1	Security Audit Data Generation
Hierarchical to:	No other components.
Dependencies:	None.

FAU_GEN_EXT.1.1. The TSF shall be able to generate audit records for each TOE component. The audit records generated by the TSF of each TOE component shall include the subset of security relevant audit events which can occur on the TOE component.

C.1.2 Protected Audit Event Storage (FAU_STG_EXT)

Family Behaviour

This component defines the requirements for the TSF to be able to securely transmit audit data between the TOE and an external IT entity.

Component levelling

FAU_STG_EXT.1 Protected audit event storage requires the TSF to use a trusted channel implementing a secure protocol.

FAU_STG_EXT.2 Counting lost audit data requires the TSF to provide information about audit records affected when the audit log becomes full.

FAU_STG_EXT.3 Action in case of possible audit data loss requires the TSF to generate a warning before the audit trail exceeds the local storage capacity.

FAU_STG_EXT.4 Protected Local audit event storage for distributed TOEs requires the TSF to use a trusted channel to protect audit transfer to another TOE component.

FAU_STG_EXT.5 Protected Remote audit event storage for distributed TOEs requires the TSF to use a trusted channel to protect audit transfer to another TOE component.

Management: FAU_STG_EXT.1, FAU_STG_EXT.2, FAU_STG_EXT.3, FAU_STG_EXT.4, FAU_STG_EXT.5

The following actions could be considered for the management functions in FMT:

a) The TSF shall have the ability to configure the cryptographic functionality.

Audit: FAU_STG_EXT.1, FAU_STG_EXT.2, FAU_STG_EXT.3, FAU_STG_EXT.4. FAU_STG_EXT.5

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) No audit necessary.

C.1.2.1 FAU_ STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1	Protected Audit Event Storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FTP_ITC.1 Inter-TSF Trusted Channel

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 [selection:

- *The TOE shall consist of a single standalone component that stores audit data locally,*
- *The TOE shall be a distributed TOE that stores audit data on the following TOE components: [assignment: identification of TOE components],*
- *The TOE shall be a distributed TOE with storage of audit data provided externally for the following TOE components: [assignment: list of TOE components that do not store audit data locally and the other TOE components to which they transmit their generated audit data].*

FAU_STG_EXT.1.3 The TSF shall [selection: *drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]*] when the local storage space for audit data is full.

C.1.2.2 FAU_ STG_EXT.2 Counting Lost Audit Data

FAU_STG_EXT.2	Counting Lost Audit Data

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.2.1 The TSF shall provide information about the number of [selection: *dropped, overwritten, [assignment: other information]*] audit records in the case where the local storage has been filled and the TSF takes one of the actions defined in FAU_STG_EXT.1.3.

C.1.2.3 FAU_ STG_EXT.3 Action in Case of Possible Audit Data Loss

FAU_STG_EXT.3	Action in Case of Possible Audit Data Loss

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.3.1/LocSpace The TSF shall *generate a warning to inform the Administrator* before the audit trail *exceeds the local audit trail storage capacity*.

C.1.2.4 FAU_ STG_EXT.4 Protected Local Audit Event Storage for Distributed TOEs

FAU_STG_EXT.4	Protected Audit Event Storage

Hierarchical to: No other components.

Dependencies: FAU_GEN_EXT.1 Security Audit data generation for Distributed TOE Components
[FPT_ITT.1 Intra-TSF Trusted Channel or
FTP_ITC.1 Inter-TSF Trusted Channel]

FAU_STG_EXT.4.1 The TSF of each TOE component which stores security audit data locally shall perform the following actions when the local storage space for audit data is full: *[assignment: table of components and for each component its action chosen according to the following:* [selection: *drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]*]].

C.1.2.5 FAU_ STG_EXT.5 Protected Remote Audit Event Storage for Distributed TOEs

FAU_STG_EXT.5	Protected Audit Event Storage

Hierarchical to: No other components.

Dependencies: FAU_GEN_EXT.1 Security Audit data generation for Distributed TOE Components
[FPT_ITT.1 Intra-TSF Trusted Channel or
FTP_ITC.1 Inter-TSF Trusted Channel]

FAU_STG_EXT.5.1 Each TOE component which does not store security audit data locally shall be able to buffer security audit data locally until it has been transferred to another TOE component that stores or forwards it. All transfer of audit records between TOE components shall use a protected channel according to [selection*: FPT_ITT.1, FTP_ITC.1*].

C.2 Cryptographic Support (FCS)

C.2.1 Random Bit Generation (FCS_RBG_EXT)

C.2.1.1 FCS_RBG_EXT.1 Random Bit Generation

Family Behaviour

Components in this family address the requirements for random bit/number generation. This is a new family defined for the FCS class.

Component levelling

FCS_RBG_EXT.1 Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source.

Management: FCS_RBG_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen

Audit: FCS_RBG_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: failure of the randomization process

FCS_RBG_EXT.1	**Random Bit Generation**
Hierarchical to:	No other components
Dependencies:	No other components

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

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [selection: *[assignment: number of software-based sources] software-based noise source, [assignment: number of platform-based sources] platform-based noise source*] with a minimum of [selection: *128 bits, 192 bits, 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.

C.2.2 Cryptographic Protocols (FCS_DTLSC_EXT, FCS_DTLSS_EXT, FCS_HTTPS_EXT, FCS_IPSEC_EXT, FCS_NTP_EXT, FCS_SSHC_EXT, FCS_SSHS_EXT, FCS_TLSC_EXT, FCS_TLSS_EXT)

C.2.2.1 FCS_DTLSC_EXT DTLS Client Protocol

Family Behaviour

The component in this family addresses the ability for a client to use DTLS to protect data between the client and a server using the DTLS protocol.

Component levelling

FCS_DTLSC_EXT.1 DTLS Client requires that the client side of DTLS be implemented as specified.

FCS_DTLSC_EXT.2 DTLS Client requires that the client side of the DTLS implementation include mutual authentication.

Management: FCS_DTLSC_EXT.1, FCS_DTLSC_EXT.2

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_DTLSC_EXT.1, FCS_DTLSC_EXT.2

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of DTLS session establishment
b) DTLS session establishment
c) DTLS session termination

FCS_DTLSC_EXT.1	**DTLS Client Protocol**
Hierarchical to:	No other components
Dependencies:	FCS_CKM. 1DataEncryption1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment

FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1/SigGen1SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

FCS_DTLSC_EXT.1.1 The TSF shall implement [selection: *DTLS 1.2 (RFC 6347), DTLS 1.0 (RFC 4347)*] supporting the following ciphersuites:

- *[assignment: List of optional ciphersuites and reference to RFC in which each is defined].*

FCS_DTLSC_EXT.1.2 The TSF shall verify that the presented identifier matches [selection: *the reference identifier per RFC 6125 section 6, IPv4 address in CN or SAN, IPv6 address in the CN or SAN, IPv4 address in SAN, IPv6 address in the SAN, the identifier per RFC 5280 Appendix A using [selection: id-at-commonName, id-at-countryName, id-at-dnQualifier, id-at-generationQualifier, id-at-givenName, id-at-initials, id-at-localityName, id-at-name, id-at-organizationalUnitName, id-at-organizationName, id-at-pseudonym, id-at-serialNumber, id-at-stateOrProvinceName, id-at-surname, id-at-title] and no other attribute types*].

FCS_DTLSC_EXT.1.3 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [selection:

- *Not implement any administrator override mechanism*
- *require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate*

].

FCS_DTLSC_EXT.1.4 The TSF shall [selection: *not present the Supported Elliptic Curves/Supported Groups Extension, present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups:* [selection: *secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192] and no other curves/groups*] in the Client Hello.

FCS_DTLSC_EXT.2	**DTLS Client Support for Mutual Authentication**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1/DataEncryption Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment

FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FCS_DTLSC_EXT.1 DTLS Client Protocol
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

FCS_DTLSC_EXT.2.1 The TSF shall support mutual authentication using X.509v3 certificates.

FCS_DTLSC_EXT.2.2 The TSF shall [selection: *terminate the DTLS session, silently discard the record*] if a message received contains an invalid MAC.

FCS_DTLSC_EXT.2.3 The TSF shall detect and silently discard replayed messages for:

- DTLS records previously received;
- DTLS records too old to fit in the sliding window.

C.2.2.2 FCS_DTLSS_EXT DTLS Server Protocol

Family Behaviour

The component in this family addresses the ability for a server to use DTLS to protect data between a client and the server using the DTLS protocol.

Component levelling

FCS_DTLSS_EXT.1 DTLS Server requires that the server side of TLS be implemented as specified.

FCS_DTLSS_EXT.2: DTLS Server requires that mutual authentication be included in the DTLS implementation.

Management: FCS_DTLSS_EXT.1, FCS_DTLSS_EXT.2

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_DTLSS_EXT.1, FCS_DTLSS_EXT.2

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of DTLS session establishment.
b) DTLS session establishment
c) DTLS session termination

FCS_DTLSS_EXT.1	**DTLS Server Protocol**

Hierarchical to: No other components

Dependencies:
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1//DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1//SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

FCS_DTLSS_EXT.1.1 The TSF shall implement [selection: *DTLS 1.2 (RFC 6347), DTLS 1.0 (RFC 4347)*] supporting the following ciphersuites:

- *[assignment: List of optional ciphersuites and reference to RFC in which each is defined]*

FCS_DTLSS_EXT.1.2 The TSF shall deny connections from clients requesting *[assignment: list of protocol versions]*.

FCS_DTLSS_EXT.1.3 The TSF shall not proceed with a connection handshake attempt if the DTLS Client fails validation.

FCS_DTLSS_EXT.1.4 The TSF shall perform key establishment for TLS using [selection: *RSA with key size* [selection*: 2048 bits, 3072 bits, 4096 bits], Diffie-Hellman parameters with size* [selection*: 2048 bits, 3072 bits, 4096 bits, 6144 bits, 8192 bits], Diffie-Hellman groups* [selection*: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, no other groups], ECDHE curves* [selection*: secp256r1, secp384r1, secp521r1] and no other curves*].

FCS_DTLSS_EXT.1.5 The TSF shall [selection: *terminate the DTLS session, silently discard the record*] if a message received contains an invalid MAC.

FCS_DTLSS_EXT.1.6 The TSF shall detect and silently discard replayed messages for:

- DTLS records previously received.
- DTLS Records too old to fit in the sliding window.

FCS_DTLSS_EXT.1.7 The TSF shall support [selection: *no session resumption or session tickets, session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2), session resumption based on session tickets according to RFC 5077*].

FCS_DTLSS_EXT.2	**DTLS Server Support for Mutual Authentication**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification) FCS_COP.1/Hash Cryptographic operation (Hash Algorithm) FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation FCS_DTLSS_EXT.1 DTLS Server Protocol

FCS_DTLSS_EXT.2.1 The TSF shall support mutual authentication of DTLS clients using X.509v3 certificates.

FCS_DTLSS_EXT.2.2 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [selection:

- *Not implement any administrator override mechanism*
- *require administrator authorization to establish the connection if the TSF fails to* [selection: *match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented client certificate*

].

FCS_DTLSS_EXT.2.3 The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the client.

C.2.2.3FCS_HTTPS_EXT.1 HTTPS Protocol

Family Behaviour

Components in this family define the requirements for protecting remote management sessions between the TOE and a Security Administrator. This family describes how HTTPS will be implemented. This is a new family defined for the FCS Class.

Component levelling

FCS_HTTPS_EXT.1 HTTPS requires that HTTPS be implemented according to RFC 2818 and supports TLS.

Management: FCS_HTTPS_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_HTTPS_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) There are no auditable events foreseen.

FCS_HTTPS_EXT.1	**HTTPS Protocol**
Hierarchical to:	No other components
Dependencies:	[FCS_TLSC_EXT.1 TLS Client Protocol, or FCS_TLSS_EXT.1 TLS Server 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 the HTTPS protocol using TLS.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [selection: *not establish the connection, request authorization to establish the connection, [assignment: other action]*] if the peer certificate is deemed invalid.

C.2.2.4 FCS_IPSEC_EXT.1 IPsec Protocol

Family Behaviour

Components in this family address the requirements for protecting communications using IPsec.

Component levelling

FCS_IPSEC_EXT.1 IPsec requires that IPsec be implemented as specified.

Management: FCS_IPSEC_EXT.1

The following actions could be considered for the management functions in FMT:

a) Maintenance of SA lifetime configuration

Audit: FCS_IPSEC_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Decisions to DISCARD, BYPASS, PROTECT network packets processed by the TOE.
b) Failure to establish an IPsec SA
c) IPsec SA establishment
d) IPsec SA termination
e) Negotiation "down" from an IKEv2 to IKEv1 exchange.

FCS_IPSEC_EXT.1	**Internet Protocol Security (IPsec) Communications**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification) FCS_COP.1/Hash Cryptographic operation (Hash Algorithm) FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation

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 [selection: *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 [selection: *AES-CBC-128 (RFC 3602), AES-CBC-192 (RFC 3602), AES-CBC-256 (RFC 3602), AES-GCM-128 (RFC 4106), AES-GCM-192 (RFC 4106), AES-GCM-256 (RFC 4106),*] together with a Secure Hash Algorithm (SHA)-based HMAC [selection: *HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, no HMAC algorithm*].

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [selection:

- *IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109,* [selection: *no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers*], *and* [selection: *no other RFCs for hash functions, RFC 4868 for hash functions*]*;*

- *IKEv2 as defined in RFCs 5996 [selection: with no support for NAT traversal, with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)*], *and* [selection: *no other RFCs for hash functions, RFC 4868 for hash functions*]].

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [selection: *IKEv1, IKEv2*] protocol uses the cryptographic algorithms [selection: *AES-CBC-128, AES_CBC-192 AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES-GCM-192, AES-GCM-256 (specified in RFC 5282)*].

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [selection:

- *IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on* [selection:
    - *number of bytes;*
    - *length of time, where the time values can be configured within [assignment: integer range including 24*] *hours;*

  ]*;*
- *IKEv2 SA lifetimes can be configured by a Security Administrator based on* [selection:
    - *number of bytes;*
    - *length of time, where the time values can be configured within [assignment: integer range including 24*] *hours*

  ]

].

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [selection:

- *IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on* [selection:
    - *number of bytes;*
    - *length of time, where the time values can be configured within [assignment: integer range including 8] hours;*

];

- *IKEv2 Child SA lifetimes can be configured by a Security Administrator based on* [selection:
  - *number of bytes;*
  - *length of time, where the time values can be configured within [assignment: integer range including 8] hours;*

]

].

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 *[assignment: (one or more) number(s) of bits that is at least twice the security strength of the negotiated Diffie-Hellman group*] bits.

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

- *according to the security strength associated with the negotiated Diffie-Hellman group;*
- *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 Group(s) [selection:

- [selection*: 14 (2048-bit MODP), 15 (3072-bit MODP), 16 (4096-bit MODP), 17 (6144-bit MODP), 18 (8192-bit MODP)] according to RFC 3526,*
- [selection*: 19 (256-bit Random ECP), 20 (384-bit Random ECP), 21 (521-bit Random ECP), 24 (2048-bit MODP with 256-bit POS)] according to RFC 5114*.

].

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 [selection: *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 [selection: *IKEv1 Phase 2, IKEv2 CHILD_SA*] connection.

FCS_IPSEC_EXT.1.13 The TSF shall ensure that all IKE protocols perform peer authentication using [selection: *RSA, ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [selection: *Pre-shared Keys, no other method*].

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: [selection: *SAN: IP address, SAN: Fully Qualified Domain Name (FQDN), SAN: user FQDN, CN: IP address, CN:*

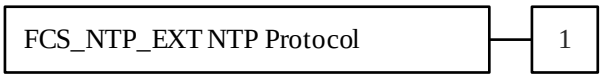
Fully Qualified Domain Name (FQDN), CN: user FQDN, Distinguished Name (DN)] and [selection: *no other reference identifier type, [assignment: other supported reference identifier types*]].

C.2.2.5 FCS_NTP_EXT.1 NTP Protocol

Family Behaviour

The component in this family addresses the ability for a TOE to protect NTP time synchronization traffic.

Component levelling



FCS_NTP_EXT.1 Requires NTP to be implemented as specified

Management: FCS_NTP_EXT.1

The following actions could be considered for the management functions in FMT:

a) Ability to configure NTP

Audit: FCS_NTP_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) No audit requirements are specified.

FCS_NTP_EXT.1	**NTP Protocol**
Hierarchical to:	No other components
Dependencies:	FCS_COP.1 Cryptographic operation [FCS_DTLSC_EXT.1 DTLSC Client Protocol or FCS_IPSEC_EXT.1 IPsec Protocol]

FCS_NTP_EXT.1.1 The TSF shall use only the following NTP version(s) [selection: *NTP v3 (RFC 1305), NTP v4 (RFC 5905)*].

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

- Authentication using [selection: *SHA1, SHA256, SHA384, SHA512, AES-CBC-128, AES-CBC-256*] as the message digest algorithm(s);

- [selection: *IPsec, DTLS*] to provide trusted communication between itself and an NTP time source.

].

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.

C.2.2.6 FCS_SSHC_EXT.1 SSH Client

Family Behaviour

The component in this family addresses the ability for a client to use SSH to protect data between the client and a server using the SSH protocol.

Component levelling

FCS_SSHC_EXT.1 SSH Client requires that the client side of SSH be implemented as specified.

Management: FCS_SSHC_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_SSHC_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of SSH session establishment
b) SSH session establishment
c) SSH session termination

FCS_SSHC_EXT.1	**SSH Client Protocol**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)

FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_SSHC_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs *4251, 4252, 4253, 4254,* [selection: *4256, 4344, 5647, 5656, 6187, 6668, 8268, 8308 section 3.1, 8332*].

FCS_SSHC_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [selection: *password-based, no other method*].

FCS_SSHC_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than *[assignment: number of bytes]* bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [assignment: *list of encryption algorithms*].

FCS_SSHC_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [selection: *ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256*] as its public key algorithm(s) and rejects all other public key algorithms

FCS_SSHC_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses *[assignment: list of data integrity MAC algorithms]* as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.7 The TSF shall ensure that *[assignment: list of key exchange methods]* are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.8 The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

FCS_SSHC_EXT.1.9 The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key and [selection: *a list of trusted certification authorities, no other methods*] as described in RFC 4251 section 4.1.

C.2.2.7 FCS_SSHS_EXT.1 SSH Server Protocol

Family Behaviour

The component in this family addresses the ability for a server to offer SSH to protect data between a client and the server using the SSH protocol.

Component levelling

FCS_SSHS_EXT.1 SSH Server requires that the server side of SSH be implemented as specified.

Management: FCS_SSHS_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_SSHS_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of SSH session establishment
b) SSH session establishment
c) SSH session termination

FCS_SSHS_EXT.1	**SSH Server Protocol**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification) FCS_COP.1/Hash Cryptographic operation (Hash Algorithm) FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [selection: *4256, 4344, 5647, 5656, 6187, 6668, 8268, 8308* section 3.1, 8332].

FCS_SSHS_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [selection: *password-based, no other method*].

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than *[assignment: number of bytes]* bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: *[assignment: encryption algorithms]*.

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [selection: *ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256*] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses *[assignment: list of MAC algorithms]* as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that *[assignment: list of key exchange methods]* are the only allowed key exchange methods used for the SSH protocol.

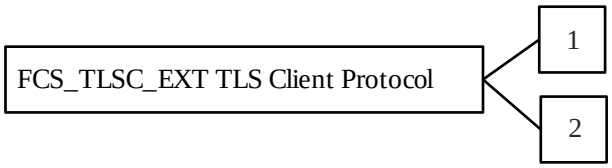
FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

C.2.2.8 FCS_TLSC_EXT TLS Client Protocol

Family Behaviour

The component in this family addresses the ability for a client to use TLS to protect data between the client and a server using the TLS protocol.

Component levelling



FCS_TLSC_EXT.1 TLS Client requires that the client side of TLS be implemented as specified.

FCS_TLSC_EXT.2 TLS Client requires that the client side of the TLS implementation include mutual authentication.

Management: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of TLS session establishment
b) TLS session establishment
c) TLS session termination

FCS_TLSC_EXT.1	**TLS Client Protocol without Mutual Authentication**

Hierarchical to: No other components

Dependencies: FCS_CKM. 1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

FCS_TLSC_EXT.1.1 The TSF shall implement [selection: *TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

- *[assignment: list of optional ciphersuites and reference to RFC in which each is defined]* and no other ciphersuites.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches [selection: *the reference identifier per RFC 6125 section 6, IPv4 address in CN or SAN, IPv6 address in the CN or SAN, IPv4 address in SAN, IPv6 address in the SAN, the identifier per RFC 5280 Appendix A using [selection: id-at-commonName, id-at-countryName, id-at-dnQualifier, id-at-generationQualifier, id-at-givenName, id-at-initials, id-at-localityName, id-at-name, id-at-organizationalUnitName, id-at-organizationName, id-at-pseudonym, id-at-serialNumber, id-at-stateOrProvinceName, id-at-surname, id-at-title] and no other attribute types*].

FCS_TLSC_EXT.1.3 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the server certificate is invalid. The TSF shall also [selection:

- *Not implement any administrator override mechanism*

- *require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate*

].

FCS_TLSC_EXT.1.4 The TSF shall [selection: *not present the Supported Elliptic Curves/Supported Groups Extension, present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups:* [selection*: secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192] and no other curves/groups*] in the Client Hello.

FCS_TLSC_EXT.2	**TLS Client Support for Mutual Authentication**

Hierarchical to: No other components

Dependencies:
- FCS_CKM.1Cryptographic Key Generation
- FCS_CKM.2 Cryptographic Key Establishment
- FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
- FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
- FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
- FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
- FCS_RBG_EXT.1 Random Bit Generation
- FCS_TLSC_EXT.1 TLS Client Protocol without mutual authentication
- FIA_X509_EXT.1 X.509 Certificate Validation
- FIA_X509_EXT.2 X.509 Certificate Authentication

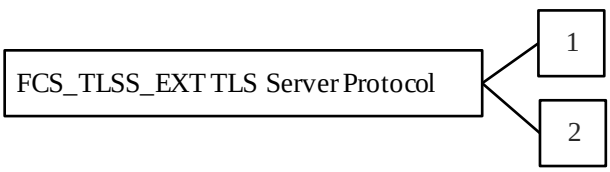
FCS_TLSC_EXT.2.1 The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

C.2.2.9 FCS_TLSS_EXT TLS Server Protocol

Family Behaviour

The component in this family addresses the ability for a server to use TLS to protect data between a client and the server using the TLS protocol.

Component levelling



FCS_TLSS_EXT.1 TLS Server requires that the server side of TLS be implemented as specified.

FCS_TLSS_EXT.2: TLS Server requires the mutual authentication be included in the TLS implementation.

Management: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of TLS session establishment
b) TLS session establishment
c) TLS session termination

FCS_TLSS_EXT.1	**TLS Server Protocol without Mutual Authentication**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification) FCS_COP.1/Hash Cryptographic operation (Hash Algorithm) FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation FIA_X509_EXT.1 X.509 Certificate Validation FIA_X509_EXT.2 X.509 Certificate Authentication

FCS_TLSS_EXT.1.1 The TSF shall implement [selection: *TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)*] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites:

- *[assignment: list of optional ciphersuites and reference to RFC in which each is defined]* and no other ciphersuites.

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [selection: *TLS 1.1, TLS 1.2, none*].

FCS_TLSS_EXT.1.3 The TSF shall perform key establishment for TLS using [selection: *RSA with key size* [selection*: 2048 bits, 3072 bits, 4096 bits], Diffie-Hellman parameters with size* [selection*: 2048 bits, 3072 bits, 4096 bits, 6144 bits, 8192 bits], Diffie-Hellman groups*

[selection: *ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, no other groups], ECDHE curves* [selection: *secp256r1, secp384r1, secp521r1] and no other curves*]].

FCS_TLSS_EXT.1.4 The TSF shall support [selection: *no session resumption or session tickets, session resumption based on session IDs according to* RFC 4346 *(TLS1.1) or* RFC 5246 *(TLS1.2), session resumption based on session tickets according to* RFC 5077].

FCS_TLSS_EXT.2	**TLS Server Support for Mutual Authentication**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification) FCS_COP.1/Hash Cryptographic operation (Hash Algorithm) FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation FCS_TLSS_EXT.1 TLS Server Protocol without mutual authentication FIA_X509_EXT.1 X.509 Certificate Validation FIA_X509_EXT.2 X.509 Certificate Authentication

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

FCS_TLSS_EXT.2.2 When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [selection:

- *Not implement any administrator override mechanism*
- *require administrator authorization to establish the connection if the TSF fails to* [selection: *match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented client certificate*

].

FCS_TLSS_EXT.2.3 The TSF shall not establish a trusted channel if the identifier contained in a certificate does not match an expected identifier for the client. If the identifier is a Fully Qualified Domain Name (FQDN), then the TSF shall match the identifiers according to RFC 6125, otherwise the TSF shall parse the identifier from the certificate and match the identifier against the expected identifier of the client as described in the TSS.

C.3 Identification and Authentication (FIA)

C.3.1 Password Management (FIA_PMG_EXT)

Family Behaviour

The TOE defines the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained.

Component levelling

FIA_PMG_EXT.1 Password management requires the TSF to support passwords with varying composition requirements, minimum lengths, maximum lifetime, and similarity constraints.

Management: FIA_PMG_EXT.1

No management functions.

Audit: FIA_PMG_EXT.1

No specific audit requirements.

C.3.1.1 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1	Password Management

Hierarchical to: No other components.

Dependencies: No other components.

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: [selection: *"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [assignment: other characters]*];

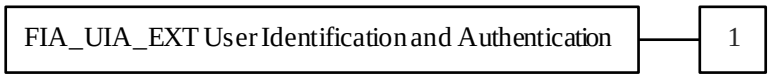
b) Minimum password length shall be configurable to between [*assignment: minimum number of characters supported by the TOE*] and [*assignment: number of characters greater than or equal to 15*] characters.

C.3.2 User Identification and Authentication (FIA_UIA_EXT)

Family Behaviour

The TSF allows certain specified actions before the non-TOE entity goes through the identification and authentication process.

Component levelling



FIA_UIA_EXT.1 User Identification and Authentication requires Administrators (including remote Administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path. It also ensures that every user is identified and authenticated before the TOE performs any mediated functions

Management: FIA_UIA_EXT.1

The following actions could be considered for the management functions in FMT:

a) Ability to configure the list of TOE services available before an entity is identified and authenticated

Audit: FIA_UIA_EXT.N

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) All use of the identification and authentication mechanism
b) Provided user identity, origin of the attempt (e.g. IP address)

C.3.2.1 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1	User Identification and Authentication
Hierarchical to:	No other components.
Dependencies:	FTA_TAB.1 Default TOE Access Banners

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;
- [selection: *no other actions, automated generation of cryptographic keys, [assignment: list of services, actions performed by the TSF in response to non-TOE requests]*].

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.

C.3.3 User authentication (FIA_UAU_EXT)

Family Behaviour

Provides for a locally based administrative user authentication mechanism

Component levelling

FIA_UAU_EXT.2 The password-based authentication mechanism provides administrative users a locally based authentication mechanism.

Management: FIA_UAU_EXT.2

The following actions could be considered for the management functions in FMT:

a) None

Audit: FIA_UAU_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: All use of the authentication mechanism

C.3.3.1 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2	**Password-based Authentication Mechanism**
Hierarchical to:	No other components.
Dependencies:	No other components.

FIA_UAU_EXT.2.1 The TSF shall provide a local [selection: *password-based, SSH public key-based, certificate-based, [assignment: other authentication mechanism(s)]*] authentication mechanism to perform local administrative user authentication.

C.3.4 Authentication using X.509 certificates (FIA_X509_EXT)

Family Behaviour

This family defines the behaviour, management, and use of X.509 certificates for functions to be performed by the TSF. Components in this family require validation of certificates according to a specified set of rules, use of certificates for authentication for protocols and integrity verification, and the generation of certificate requests.

Component levelling

FIA_X509_EXT.1 X509 Certificate Validation, requires the TSF to check and validate certificates in accordance with the RFCs and rules specified in the component.

FIA_X509_EXT.2 X509 Certificate Authentication, requires the TSF to use certificates to authenticate peers in protocols that support certificates, as well as for integrity verification and potentially other functions that require certificates.

FIA_X509_EXT.3 X509 Certificate Requests, requires the TSF to be able to generate Certificate Request Messages and validate responses.

Management: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

The following actions could be considered for the management functions in FMT:

a) Remove imported X.509v3 certificates
b) Approve import and removal of X.509v3 certificates
c) Initiate certificate requests

Audit: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: No specific audit requirements are specified.

C.3.4.1 FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1	**X.509 Certificate Validation**

Hierarchical to: No other components

Dependencies: FIA_X509_EXT.2 X.509 Certificate Authentication

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

- RFC 5280 certificate validation and certification path validation.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.

- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.

- The TSF shall validate the revocation status of the certificate using [selection: *the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5, no revocation method*]

- The TSF shall validate the extendedKeyUsage field according to the following rules: *[assignment: rules that govern contents of the extendedKeyUsage field that need to be verified]*.

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

C.3.4.2 FIA_X509_EXT.2 X509 Certificate Authentication

FIA_X509_EXT.2	**X.509 Certificate Authentication**
Hierarchical to:	No other components
Dependencies:	FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection*: DTLS, HTTPS, IPsec, TLS, SSH, [assignment: other protocols], no protocols*], and [selection: *code signing for system software updates [assignment: other uses], no additional uses*].

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [selection: *allow the Administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

C.3.4.3FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3	**X.509 Certificate Requests**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FIA_X509_EXT.1 X.509 Certificate Validation

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 [selection: *device-specific information, Common Name, Organization, Organizational Unit, Country, [assignment: other information]*].

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

C.4 Protection of the TSF (FPT)

C.4.1 Protection of TSF Data (FPT_SKP_EXT)

Family Behaviour

Components in this family address the requirements for managing and protecting TSF data, such as cryptographic keys. This is a new family modelled after the FPT_PTD Class.

Component levelling

FPT_SKP_EXT.1 Protection of TSF Data (for reading all symmetric keys), requires preventing symmetric keys from being read by any user or subject. It is the only component of this family.

Management: FPT_SKP_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) There are no auditable events foreseen.

C.4.1.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

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

Hierarchical to: No other components.

Dependencies: No other components.

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

C.4.2 Protection of Administrator Passwords (FPT_APW_EXT)

C.4.2.1 FPT_APW_EXT.1 Protection of Administrator Passwords

Family Behaviour

Components in this family ensure that the TSF will protect plaintext credential data such as passwords from unauthorized disclosure.

Component levelling

FPT_APW_EXT.1 Protection of Administrator passwords requires that the TSF prevent plaintext credential data from being read by any user or subject.

Management: FPT_APW_EXT.1

The following actions could be considered for the management functions in FMT:

a) No management functions.

Audit: FPT_APW_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) No audit necessary.

FPT_APW_EXT.1	**Protection of Administrator Passwords**
Hierarchical to:	No other components
Dependencies:	No other components.

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.

C.4.3 TSF Self-Test (FPT_TST_EXT)

C.4.3.1 FPT_TST_EXT.1 TSF Testing

Family Behaviour

Components in this family address the requirements for self-testing the TSF for selected correct operation.

Component levelling

FPT_TST_EXT.1 TSF Self-Test requires a suite of self-tests to be run during initial start-up in order to demonstrate correct operation of the TSF.

Management: FPT_TST_EXT.1

The following actions could be considered for the management functions in FMT:

a) No management functions.

Audit: FPT_TST_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Indication that TSF self-test was completed
b) Failure of self-test

FPT_TST_EXT.1	**TSF Testing**
Hierarchical to:	No other components.
Dependencies:	No other components.

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [selection: *during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]*] to demonstrate the correct operation of the TSF: *[assignment: list of self-tests run by the TSF*].

C.4.4 Trusted Update (FPT_TUD_EXT)

Family Behaviour

Components in this family address the requirements for updating the TOE firmware and/or software.

Component levelling

FPT_TUD_EXT.1 Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.

FPT_TUD_EXT.2 Trusted update based on certificates applies when using certificates as part of trusted update and requires that the update does not install if a certificate is invalid.

Management: FPT_TUD_EXT.1, FPT_TUD_EXT.2

The following actions could be considered for the management functions in FMT:

a) Ability to update the TOE and to verify the updates
b) Ability to update the TOE and to verify the updates using the digital signature capability (FCS_COP.1/SigGen) and [selection: *no other functions, [assignment: other cryptographic functions (or other functions) used to support the update capability]*]
c) Ability to update the TOE, and to verify the updates using [selection: *digital signature, published hash, no other mechanism*] capability prior to installing those updates

Audit: FPT_TUD_EXT.1, FPT_TUD_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Initiation of the update process.
b) Any failure to verify the integrity of the update

C.4.4.1 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1	**Trusted Update**
Hierarchical to:	No other components
Dependencies:	FCS_COP.1/SigGen Cryptographic operation (for Cryptographic Signature and Verification), or FCS_COP.1/Hash Cryptographic operation (for cryptographic hashing)

FPT_TUD_EXT.1.1 The TSF shall provide [assignment: *Administrators*] the ability to query the currently executing version of the TOE firmware/software and [selection: *the most recently installed version of the TOE firmware/software; no other TOE firmware/software version*].

FPT_TUD_EXT.1.2 The TSF shall provide [assignment: *Administrators*] the ability to manually initiate updates to TOE firmware/software and [selection: *support automatic checking for updates, support automatic updates, no other update mechanism*].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [selection: *X.509 certificate, digital signature, published hash*] prior to installing those updates.

C.4.4.2 FPT_TUD_EXT.2 Trusted Update Based on Certificates

FPT_TUD_EXT.2	Trusted Update Based on Certificates

Hierarchical to: No other components

Dependencies: FPT_TUD_EXT.1

FPT_TUD_EXT.2.1 The TSF shall check the validity of the code signing certificate before installing each update.

FPT_TUD_EXT.2.2 If revocation information is not available for a certificate in the trust chain that is not a trusted certificate designated as a trust anchor, the TSF shall [selection: *not install the update, allow the Administrator to choose whether to accept the certificate in these cases*].

FPT_TUD_EXT.2.3 If the certificate is deemed invalid because the certificate has expired, the TSF shall [selection: *allow the Administrator to choose whether to install the update in these cases, not accept the certificate*].

FPT_TUD_EXT.2.4 If the certificate is deemed invalid for reasons other than expiration or revocation information being unavailable, the TSF shall not install the update.

C.4.5 Time stamps (FPT_STM_EXT)

Family Behaviour

Components in this family extend FPT_STM requirements by describing the source of time used in timestamps.

Component levelling

FPT_STM_EXT.1 Reliable Time Stamps is hierarchic to FPT_STM.1: it requires that the TSF provide reliable time stamps for TSF and identifies the source of the time used in those timestamps.

Management: FPT_STM_EXT.1

The following actions could be considered for the management functions in FMT:

a) Management of the time
b) Administrator setting of the time.

Audit: FTA_SSL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Discontinuous changes to the time.

C.4.5.1FPT_STM_EXT.1 Reliable Time Stamps

FPT_STM_EXT.1	Reliable Time Stamps

Hierarchical to: No other components

Dependencies: No other components.

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 [selection: *allow the Security Administrator to set the time, synchronise time with an NTP server*].

C.5 TOE Access (FTA)

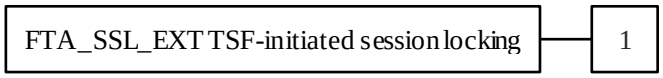
C.5.1 TSF-initiated Session Locking (FTA_SSL_EXT)

Family Behaviour

Components in this family address the requirements for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

The extended FTA_SSL_EXT family is based on the FTA_SSL family.

Component levelling



FTA_SSL_EXT.1 TSF-initiated session locking, requires system initiated locking of an interactive session after a specified period of inactivity. It is the only component of this family.

Management: FTA_SSL_EXT.1

The following actions could be considered for the management functions in FMT:

c) Specification of the time of user inactivity after which lock-out occurs for an individual user.

Audit: FTA_SSL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

b) Any attempts at unlocking an interactive session.

C.5.1.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1	TSF-initiated Session Locking

Hierarchical to: No other components

Dependencies: FIA_UAU.1 Timing of authentication

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

- *lock the session - disable any activity of the Administrator's data access/display devices other than unlocking the session, and requiring that the Administrator re-authenticate to the TSF prior to unlocking the session;*
- *terminate the session*]

after a Security Administrator-specified time period of inactivity.

C.6 Communication (FCO)

C.6.1 Communication Partner Control (FCO_CPC_EXT)

Family Behaviour

This family is used to define high-level constraints on the ways that partner IT entities communicate. For example, there may be constraints on when communication channels can be used, how they are established, and links to SFRs expressing lower-level security properties of the channels.

Component levelling

FCO_CPC_EXT.1 Component Registration Channel Definition, requires the TSF to support a registration channel for joining together components of a distributed TOE, and to ensure that the availability of this channel is under the control of an Administrator. It also requires statement of the type of channel used (allowing specification of further lower-level security requirements by reference to other SFRs).

Management: FCO_CPC_EXT.1

No separate management functions are required. Note that elements of the SFR already specify certain constraints on communication in order to ensure that the process of forming a distributed TOE is a controlled activity.

Audit: FCO_CPC_EXT.1

The following actions should be auditable if FCO_CPC_EXT.1 is included in the PP/ST:

a) Enabling communications between a pair of components as in FCO_CPC_EXT.1.1 (including identities of the endpoints).
b) Disabling communications between a pair of components as in FCO_CPC_EXT.1.3 (including identity of the endpoint that is disabled).

If the required types of channel in FCO_CPC_EXT.1.2 are specified by using other SFRs then the use of the registration channel may be sufficiently covered by the audit requirements on those SFRs: otherwise a separate audit requirement to audit the use of the channel should be identified for FCO_CPC_EXT.1.

C.6.1.1FCO_CPC_EXT.1 Component Registration Channel Definition

FCO_CPC_EXT.1	**Component Registration Channel Definition**

Hierarchical to: No other components.

Dependencies: No other components.

FCO_CPC_EXT.1.1 The TSF shall require a Security Administrator to enable communications between any pair of TOE components before such communication can take place.

FCO_CPC_EXT.1.2 The TSF shall implement a registration process in which components establish and use a communications channel that uses [assignment: *list of different types of channel given in the form of a selection*] for at least [assignment: *type of data for which the channel must be used*].

FCO_CPC_EXT.1.3 The TSF shall enable a Security Administrator to disable communications between any pair of TOE components.