FortiGate 6000 Series w/ FortiOS 5.6

Security Target

Version 1.3

July 2020

Document prepared by

Lightship Security

www.lightshipsec.com

Document History

Version	Date	Author	Description
1.0	4 Jun 2020	L Turner	Release for certification.
1.1	30 Jun 2020	L Turner	Address observations.
1.2	15 Jul 2020	L Turner	Update guidance reference.
1.3	28 Jul 2020	L Turner	Update TOE name.

Table of Contents

List of Tables

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the Fortinet FortiGate 6000 Series w/ FortiOS 5.6 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 6000 Series w/ FortiOS 5.6 Version 5.6.6 Build 4265
Security Target	FortiGate 6000 Series w/ FortiOS 5.6 Security Target, v1.3

1.3 Conformance Claims

3 This ST supports the following conformance claims:

a) CC version 3.1 revision 4

b) CC Part 2 extended

c) CC Part 3 conformant

d) collaborative Protection Profile for Stateful Traffic Filter Firewalls (FWcPP), Version 2.0 + Errata 20180314

e) Network Device collaborative Protection Profile Extended Package - VPN Gateway (VPN_EP), Version 2.1

f) collaborative Protection Profile for Network Devices/collaborative Protection Profile for Stateful Traffic Filter Firewalls Extended Package for Intrusion Prevention Systems (IPS_EP), Version 2.11

g) NIAP Technical Decisions per Table 2

Table 2: NIAP Technical Decisions

TD #	PP/EP	Name	Rationale if n/a
TD0209	VPN_EP	Additional DH Group added as selection for IKE Protocols	
TD0242	VPN_EP	FPF_RUL_EXT.1.7, Test 3 - Logging Dropped Packets	
TD0248	VPN_EP	FAU_GEN.1 Guidance Activity	
TD0259	FWcPP	NIT Technical Decision for Support for X509 ssh rsa authentication IAW RFC 6187	

TD #	PP/EP	Name	Rationale if n/a
TD0291	FWcPP	NIT technical decision for DH14 and FCS_CKM.1	
TD0307	VPN_EP	Modification of FTP_ITC_EXT.1.1	
TD0316	VPN_EP	Update to FPT_TST_EXT.2.1	
TD0317	VPN_EP	FMT_MOF.1/Services and FMT_MTD.1/CryptoKeys	
TD0319	VPN_EP	Updates to FMT_SMF.1 in VPN Gateway EP	
TD0321	FWcPP	Protection of NTP communications	TOE does not use NTP
TD0325	IPS_EP	Inline mode for Signature-based IPS policies	
TD0329	VPN_EP	IPSEC X.509 Authentication Requirements	
TD0333	FWcPP	NIT Technical Decision for Applicability of FIA_X509_EXT.3	
TD0335	FWcPP	NIT Technical Decision for FCS_DTLS Mandatory Cipher Suites	FCS_DTLS not claimed.
TD0337	FWcPP	NIT Technical Decision for Selections in FCS_SSH*_EXT.1.6	
TD0339	FWcPP	NIT Technical Decision for Making password-based authentication optional in FCS_SSHS_EXT.1.2	
TD0340	FWcPP	NIT Technical Decision for Handling of the basicConstraints extension in CA and leaf certificates	
TD0343	FWcPP	NIT Technical Decision for Updating FCS_IPSEC_EXT.1.14 Tests	
TD0356	VPN_EP	OE.CONNECTIONS added to VPN GW v2.1	
TD0394	FWcPP	NIT Technical Decision for Audit of Management Activities related to Cryptographic Keys	
TD0398	FWcPP	NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR	
TD0400	FWcPP	NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment	
TD0402	FWcPP	NIT Technical Decision for RSA-based FCS_CKM.2 Selection	
TD0408	FWcPP	NIT Technical Decision for local vs. remote administrator accounts	

TD #	PP/EP	Name	Rationale if n/a
TD0411	FWcPP	NIT Technical Decision for FCS_SSHC_EXT.1.5, Test 1 - Server and client side seem to be confused	FCS_SSHC not claimed.
TD0412	FWcPP	NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy	
TD0423	FWcPP	NIT Technical Decision for Clarification about application of RfI#201726rev2	
TD0436	VPN_EP	IPsec protocol ESP algorithms	
TD0447	FWcPP	NIT Technical Decision for Using 'diffie-hellman-group-exchange-sha256' in FCS_SSHC/S_EXT.1.7	
TD0451	FWcPP	NIT Technical Decision for ITT Comm UUID Reference Identifier	
TD0453	FWcPP	NIT Technical Decision for Clarify authentication methods SSH clients can use to authenticate SSH se	FCS_SSHC not claimed.
TD0475	FWcPP	NIT Technical Decision for Separate traffic consideration for SSH rekey	
TD0476	FWcPP	NIT Technical Decision for Conflicting FW rules cannot be configured	
TD0478	FWcPP	NIT Technical Decision for Application Notes for FIA_X509_EXT.1 iterations	
TD0481	FWcPP	NIT Technical Decision for FCS_(D)TLSC_EXT.X.2 IP addresses in reference identifiers	
TD0482	FWcPP	NIT Technical Decision for Identification of usage of cryptographic schemes	
TD0483	FWcPP	NIT Technical Decision for Applicability of FPT_APW_EXT.1	
TD0484	FWcPP	NIT Technical Decision for Interactive sessions in FTA_SSL_EXT.1 & FTA_SSL.3	

1.4 Terminology

Table 3: Terminology

Term	Definition
BGP	Border Gateway Protocol
CC	Common Criteria

Term	Definition
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.

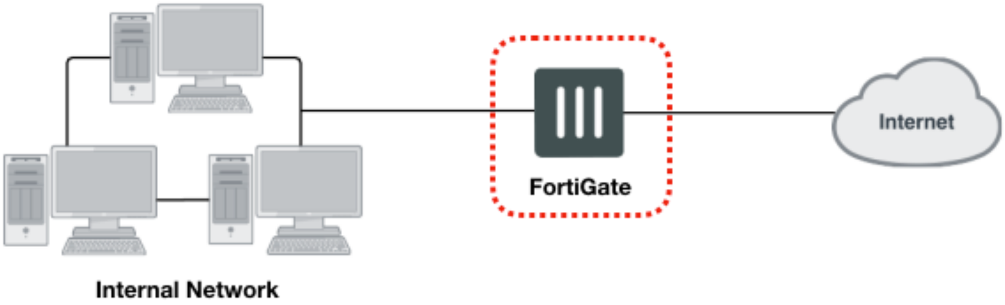


Figure 1: Example TOE deployment

2.2.2 Interfaces

6 The TOE interfaces are shown in Figure 2.

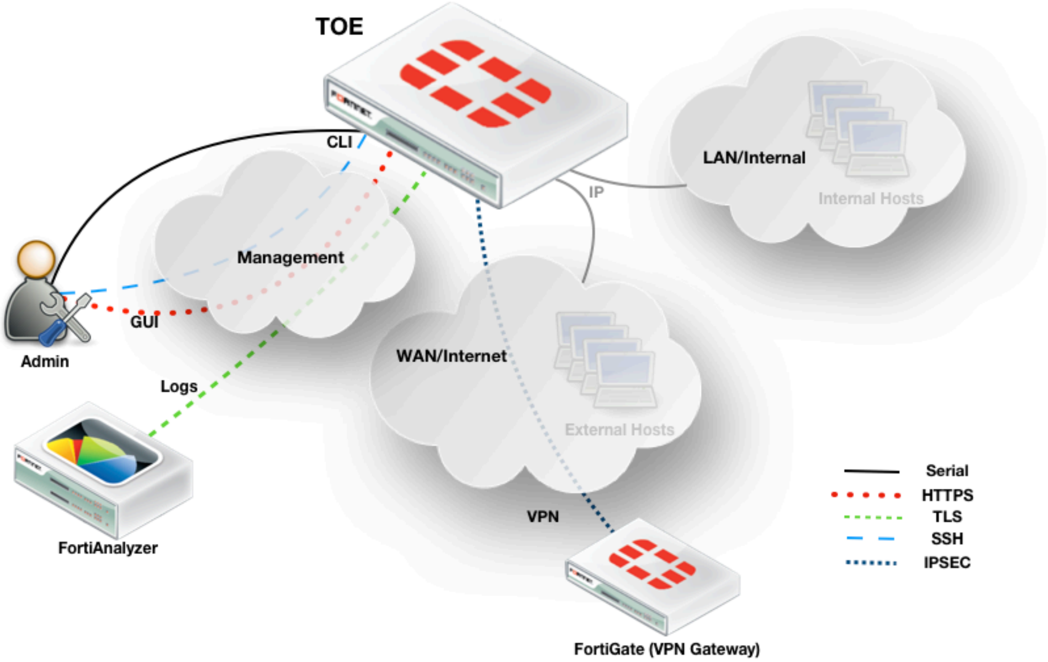


Figure 2: TOE interfaces

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 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.

2.4 Physical Scope

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

Table 4: TOE Hardware Models

Model	CPU	Architecture	RAM	Boot	Storage	ASIC
FG-6300F	Xeon D-1567	Broadwell	192GB	16GB	n/a	CP9
FG-6301F	Xeon D-1567	Broadwell	192GB	16GB	2x 1TB	CP9
FG-6500F	Xeon D-1567	Broadwell	320GB	16GB	n/a	CP9
FG-6501F	Xeon D-1567	Broadwell	320GB	16GB	2x 1TB	CP9

2.4.1 Guidance Documents

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

a) FortiOS 5.6 and FortiGate NGFW Appliances FIPS 140-2 and Common Criteria Technote, 01-567-535352-20190122, July 14, 2020

b) FortiOS Handbook - CLI Reference version 5.6.10, 01-5610-498240-20190729

c) FortiOS Log Reference Version 5.6.8, Doc No. 01-568-414447-20190131

d) FortiOS Handbook version 5.6.11, Doc No. 01-5611-497911-20190820

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

f) FortiOS Handbook – FortiGate-6000, Version 5.6.7, 01-567-465651-20190416

g) FortiGate-6000F System Guide, 01-545-464766-20180523

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

2.4.2 Non-TOE Components

12 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 IPsec VPN endpoints.

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

2.5 Logical Scope

13 The logical scope of the TOE comprises the security functions defined in section 2.3.

2.5.1 Functions not included in the TOE Evaluation

14 The FortiGate appliances are capable of a variety of functions and configurations which are not covered by the FWcPP and associated EPs. As such, 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) Use of NTP

o) Use of FortiCloud

3 Security Problem Definition

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

3.1 Threats

Table 5: Threats (FWcPP)

Identifier	Description
T.UNAUTHORIZED_ ADMINISTRATOR_ ACCESS	Threat agents may attempt to gain administrator access to the firewall by nefarious means such as masquerading as an administrator to the firewall, masquerading as the firewall 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 the firewall and a network device. Successfully gaining administrator access allows malicious actions that compromise the security functionality of the firewall 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 firewalls 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 firewall 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 firewall 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 firewall without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and

Identifier	Description
	the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_ FUNCTIONALITY_ COMPROMISE	Threat agents may compromise credentials and firewall data enabling continued access to the firewall 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 firewall 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 firewall. Having privileged access to the firewall 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.
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 6: Threats (VPN_EP)

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.

Identifier	Description
T.NETWORK_MISUSE	Access to services made available by a protected network might be used counter to Operational Environment policies.
T.REPLAY_ATTACK	If malicious or external IT entities are able to gain access to the network, they may have the ability to capture information traversing throughout the network and send them on to the intended receiver.
T.DATA_INTEGRITY	A malicious party attempts to change the data being sent – resulting in loss of integrity.

Table 7: Threats (IPS_EP)

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 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. Resource exhaustion may occur in the event of co-ordinate service request flooding from a small number of sources.

3.2 Assumptions

Table 8: Assumptions (FWcPP)

Identifier	Description
A.PHYSICAL_PROTECTION	The firewall device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the firewall's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the firewall and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the firewall that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the firewall.
A.LIMITED_ FUNCTIONALITY	The firewall device is assumed to provide networking and filtering functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example the firewall device should not provide computing platform for general purpose applications (unrelated to networking/filtering functionality).
A.TRUSTED_ ADMINSTRATOR	The Security Administrator(s) for the firewall device are assumed to be trusted and to act in the best interest of security for the organization. This includes being 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 firewall. The firewall device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the firewall device.
A.REGULAR_UPDATES	The firewall 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 firewall are protected by the host 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 firewall equipment when the equipment is discarded or removed from its operational environment.

Table 9: Assumptions (VPN_EP and IPS_EP)

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

3.3 Organizational Security Policies

Table 10: Organizational Security Policies

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.
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 11: Security Objectives for the TOE (VPN_EP)

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.

Identifier	Description
O.FAIL_SECURE	There may be instances where the TOE's hardware malfunctions or the integrity of the TOE's software is compromised, the latter being due to malicious or non-malicious intent. To address the concern of the TOE operating outside of its hardware or software specification, the TOE will shut down upon discovery of a problem reported via the self-test mechanism and provide signature-based validation of updates to the TSF.
O.PORT_FILTERING	To further address the issues associated with unauthorized disclosure of information, etc., a compliant TOE's port filtering capability will restrict the flow of network traffic between protected networks and other attached networks based on the originating (source) and/or receiving (destination) port (or service) identified in the network traffic as well as on established connection information.
O.SYSTEM_MONITORING	To address the issues of administrators being able to monitor the operations of the VPN gateway, it is necessary to provide a capability to monitor system activity. Compliant TOEs will implement the ability to log the flow of network traffic. Specifically, the TOE will provide the means for administrators to configure packet filtering rules to 'log' when network traffic is found to match the configured rule. As a result, matching a rule configured to 'log' will result in informative event logs whenever a match occurs. In addition, the establishment of security associations (SAs) is auditable, not only between peer VPN gateways, but also with certification authorities (CAs).
O.TOE_ADMINISTRATION	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 12: Security Objectives for the TOE (IPS_EP)

Identifier	Description
O.SYSTEM_MONITORING	The IPS must collect and store information about all events that may indicate an IPS policy violation related to misuse, inappropriate access, or malicious activity on monitored networks.
O.IPS_ANALYZE	The IPS must apply analytical processes to network traffic data collected from monitored networks and derive conclusions about potential intrusions or network traffic policy violations.
O.IPS_REACT	The IPS must respond appropriately to its analytical conclusions about IPS policy violations.
O.TOE_ADMINISTRATION	The IPS will provide a method for authorized administrator to configure the TSF.
O.TRUSTED_ COMMUNICATIONS	The IPS will ensure that communications between distributed components of the TOE are not subject to unauthorized modification or disclosure.

4.2 Security Objectives for the Environment

Table 13: Security Objectives for the Environment

Identifier	Description
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_ PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
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.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.
OE.CONNECTIONS (VPN_EP)	TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.

5 Security Requirements

5.1 Conventions

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 14: Summary of SFRs

Requirement	Title
FAU_GEN.1	Audit data generation
FAU_GEN.1/IPS	Audit data generation (IPS)
FAU_GEN.2	User identity association
FAU_STG_EXT.1	Security audit event storage
FCS_CKM.1	Cryptographic key generation
FCS_CKM.1/IKE	Cryptographic key generation (for IKE peer authentication)
FCS_CKM.2	Cryptographic key establishment
FCS_CKM.4	Cryptographic key destruction
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_HTTPS_EXT.1	HTTPS protocol
FCS_SSHS_EXT.1	SSH server protocol
FCS_TLSC_EXT.2	TLS Client protocol with authentication

Requirement	Title
FCS_TLSS_EXT.1	TLS Server protocol
FCS_IPSEC_EXT.1	IPsec protocol
FDP_RIP.2	Full residual information protection
FIA_AFL.1	Authentication failure handling
FIA_PMG_EXT.1	Password management
FIA_PSK_EXT.1	Pre-shared key composition
FIA_UAU_EXT.2	Password-based authentication mechanism
FIA_UAU.7	Protected authentication feedback
FIA_UIA_EXT.1	User identification and authentication
FIA_X509_EXT.1/Rev	X.509 certificate validation
FIA_X509_EXT.2	X.509 certificate authentication
FIA_X509_EXT.3	X.509 certificate requests
FMT_MOF.1/ManualUpdate	Management of security functions behaviour (Trusted Update)
FMT_MOF.1/Functions	Management of security functions behaviour
FMT_MOF.1.1/Services	Management of security functions behaviour
FMT_MTD.1/CoreData	Management of TSF data
FMT_MTD.1/CryptoKeys	Management of TSF data
FMT_SMF.1	Specification of management functions
FMT_SMF.1/IPS	Specification of management functions (IPS)
FMT_SMR.2	Restrictions on security roles
FPT_SKP_EXT.1	Protection of TSF data (for reading of all pre-shared, symmetric and private keys)
FPT_APW_EXT.1	Protection of administrator passwords
FPT_TST_EXT.1	TSF testing
FPT_TST_EXT.3	Extended: TSF testing
FPT_TUD_EXT.1	Trusted updates

Requirement	Title
FPT_STM_EXT.1	Reliable time stamps
FPT_FLS.1/SelfTest	Fail secure (Self-test failures)
FTA_SSL_EXT.1	TSF-initiated session locking
FTA_SSL.3	TSF-initiated termination
FTA_SSL.4	User-initiated termination
FTA_TAB.1	Default TOE access banners
FTP_ITC.1	Inter-TSF trusted channel
FTP_TRP.1/Admin	Trusted path
FFW_RUL_EXT.1	Stateful traffic filtering
FPF_RUL_EXT.1	Packet filtering
IPS_ABD_EXT.1	Anomaly-based IPS functionality
IPS_IPB_EXT.1	IP blocking
IPS_NTA_EXT.1	Network traffic analysis
IPS_SBD_EXT.1	Signature-based IPS functionality

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.
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).

Requirement	Auditable Events	Additional Audit Record Contents
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate	Reason for failure
	Session establishment with CA	Entire packet contents of packets transmitted/received during session establishment
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	Modification of the behaviour of the transmission of audit data to an external IT entity, the handling of audit data, the audit functionality when Local Audit Storage Space is full.	None.
FMT_MOF.1/Services	Starting and stopping of services.	None.
FMT_MTD.1/CoreData	All management activities of TSF data.	None.
FMT_MTD.1/CryptoKeys	Management of cryptographic keys.	None.
FMT_SMF.1	None.	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_TST_EXT.3	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

Requirement	Auditable Events	Additional Audit Record Contents
		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.
FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface
	Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets Identifier of rule causing packet drop
FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface
	Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets

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 **Refinement:** 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 administrative actions;~~

d) [All dissimilar IPS events;

e) All dissimilar IPS reactions;

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

g) Totals of similar reactions occurring within a specified time period.]

FAU_GEN.1.2/IPS **Refinement:** 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, *Specifically defined auditable events listed in ~~Table 4-2~~* ***the table below***.

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.

Requirement	Auditable Events	Additional Audit Record Contents
		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.

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” [selection: P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4;
- FFC Schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3

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

Application Note: The above SFR is altered by TD0291

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

FCS_CKM.1.1/IKE **Refinement:** The TSF shall generate **asymmetric** cryptographic keys **used for IKE peer authentication** in accordance with ~~a~~:

- **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:

- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1;
- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;
- Key establishment scheme using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3.

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

Application Note: The above SFR is altered by TD0402.

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 [a pseudo-random pattern using the TSF’s RBG];*
- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that logically*

addresses the storage location of the key and performs a [single] overwrite consisting of [a pseudo-random pattern using the TSF's RBG];

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] mode* and cryptographic key sizes *[128 bits, 256 bits]* that meet the following: *AES as specified in ISO 18033-3, [CBC as specified in ISO 10116, GCM as specified in ISO 19772*].

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 that complies with RFCs [4251, 4252, 4253, 4254, 6668].

Application Note: TD0398 alters the available selections for the above element.

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].

Application Note: The above element is altered by TD0339.

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].

Application Note: TD0337 altered the available selections for the above element.

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.

Application Note: TD0259 altered the available selections for the above element.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [hmac-sha1, hmac-sha2-256, hmac-sha2-512] and [no other MAC algorithms] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

Application Note: TD0337 altered the available selections for the above element.

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.

Application Note: The above element is altered by TD0475.

FCS_TLSC_EXT.2 TLS Client protocol

FCS_TLSC_EXT.2.1 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_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- 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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_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].

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifiers of the following types: [identifiers defined in RFC 6125, IPv4 address in SAN] are matched to reference identifiers.

FCS_TLSC_EXT.2.3 The TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [not establish the connection].

FCS_TLSC_EXT.2.4 The TSF shall [present the Supported Elliptic Curves Extension with the following NIST curves: [secp256r1] and no other curves] in the Client Hello.

FCS_TLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

Application Note: The above SFR is altered by TD0481

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)] supporting the following ciphersuites: [

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_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].

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 RSA key establishment with key size [2048 bits], generate EC Diffie-Hellman parameters over NIST curves [secp256r1] and no other curves; generate Diffie-Hellman parameters of size 2048 bits and [no other size]].

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, AES-CBC-256 (specified by RFC 3602)] together with a Secure Hash Algorithm (SHA)-based HMAC [HMAC-SHA-256] and [AES-GCM-128, AES-GCM-256 (specified in RFC 4106)].

Application Note: Above element is from the FWcPP with operations completed to meet the requirements of the VPN_EP.

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 [
  - 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 all IKE protocols implement DH Groups 14 (2048-bit MODP), 19 (256-bit Random ECP), 20 (384-bit Random ECP), and [no other DH groups].

Application Note: Above element is from the VPN_EP and altered by TD0209.

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)] and [no other reference identifier type].

Application Note: The above element is altered by TD0329 and TD0343.

Application Note: The IPSec SFR is the base FCS_IPSEC_EXT.1 taken from the FWcPP, augmented with the additional requirements specified in the VPN_EP

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***.**

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending remote Administrator from successfully authenticating until an Administrator defined time period has elapsed].

Application Note: This SFR combines the versions from FWcPP and VPN_EP.

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 authentication mechanism, and [no other 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;
- [*query the TOE version*]

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.
- 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.*

Application Note: The above element is altered by TD0340.

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, TLS, HTTPS], 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 [accept the certificate, not accept the certificate].

Application Note: The TOE will use the last cached information available about certificate. Therefore, the appropriate selections from FIA_X509_EXT.2.2 are “accept the certificate” as well as “not accept the certificate” depending on the last saved state.

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 [device-specific information, 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 enable and disable the functions **and services** to Security Administrators.

FMT_MTD.1/CoreData Management of TSF data

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

FMT_MTD.1/CryptoKeys Management of TSF data

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

Application Note: The above SFR is altered by TD0317.

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 configure firewall rules;*
- ***Ability to configure the cryptographic functionality,***
- ***Ability to configure the lifetime for IPsec SAs,***
- ***Ability to import X.509v3 certificates,***
- ***Ability to enable, disable, determine and modify the behavior of all the security functions of the TOE identified in this EP to the Administrator,***
- ***Ability to configure all security management functions identified in other sections of this EP**.*
- *[*
  - *Ability to configure the reference identifier for the peer*
  - Ability to configure audit behaviour;
  - Ability to set the time which is used for time-stamps;*].*

Application Note: Includes additions from the VPN_EP as altered by TD0319.

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_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_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_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.

Application Note: The above SFR is altered by TD0483.

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 Extended: TSF testing

FPT_TST_EXT.3.1 The TSF shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic service specified in FCS_COP.1/SigGen.

Application Note: From VPN_EP, SFR numbering altered by TD0316.

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.

Application note This is the base FPT_TUD_EXT.1 taken from the FWcPP, augmented by the VPN_EP.

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 use **IPsec**, and [TLS] to provide a trusted communication channel between itself **and authorized IT entities supporting the following capabilities: audit server, VPN communications**, [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*].

Application note This is the base FPT_ITC.1 taken from the FWcPP, augmented with the VPNEP. The name of the SFR is consistent with TD0307.

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

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

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

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

h) [no other rules].

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

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

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

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

FFW_RUL_EXT.1.8 The TSF shall process the applicable Stateful Traffic Filtering rules in an administratively defined order.

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

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

5.3.10 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 process the following network traffic protocols:

- Internet Protocol (IPv4)
- Internet Protocol version 6 (IPv6)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

and be capable of inspecting network packet header fields defined by the following RFCs to the extent mandated in the other elements of this SFR:

- RFC 791 (IPv4)
- RFC 2460 (IPv6)
- RFC 793 (TCP)
- RFC 768 (UDP).

FPF_RUL_EXT.1.3 The TSF shall allow the definition of Packet Filtering rules using the following network protocol fields:

- IPv4
  - Source address
  - Destination Address
  - Protocol
- IPv6
  - Source address
  - Destination Address
  - Next Header (Protocol)
- TCP
  - Source Port
  - Destination Port

- UDP
  - Source Port
  - Destination Port

and distinct interface.

FPF_RUL_EXT.1.4 The TSF shall allow the following operations to be associated with Packet Traffic Filtering rules: permit, ~~deny~~, discard, and log.

FPF_RUL_EXT.1.5 The TSF shall allow the Packet Traffic Filtering rules to be assigned to each distinct network interface.

FPF_RUL_EXT.1.6 The TSF shall process the applicable Packet Filtering rules (as determined in accordance with FPF_RUL_EXT.1.5) in the following order: Administrator-defined.

FPF_RUL_EXT.1.7 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.

IPS_IPB_EXT.1.2 The TSF shall allow IPS Administrators and [no other roles] 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 [*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]

In addition, the TSF shall support stream reassembly or equivalent to detect malicious payload even if it is split across multiple non-fragmented packets.

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]

Application Note: The above element is altered by TD0325.

5.4 Assurance Requirements

18 The TOE security assurance requirements are summarized in Table 15.

Table 15: 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 section 6.1 of the FWcPP, 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 Local log files can only be deleted via the CLI by an authorized administrator. No editing of log data is permitted.

24 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.

25 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.

26 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

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

Table 16: 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: • Pg. 64 & 65 – If an error is encountered during the generation process invalid values are returned.

Method	Key Size (bits)	Curves	Standard
Eliptic-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 17: Key Establishment Methods

Method	Usage	Service
RSA schemes	Used in TLS ciphersuites with RSA key exchange. TOE is both sender and receiver.	TLS (Audit Server) TLS/HTTPS (GUI)
Eliptic-curve schemes	Used in TLS ciphersuites with ECDH key exchange. 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.	IPsec (VPN) TLS (Audit Server) TLS/HTTPS (GUI) SSH (Admin CLI)

Table 18: 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

28 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

29 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 a random pattern.

b) **Non-volatile flash memory (Flash RAM).** The TOE performs a single, direct overwrite consisting of a random pattern.

30 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.

31 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 19: 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 random pattern when no longer needed.

Key/CSP	Storage location and method	Usage	Zeroization
IPSec Manual Encryption Key	Plaintext in RAM	Used as IPsec Session Encryption Key using AES (128-, 256-bit)	Overwritten with random pattern 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 random pattern when no longer needed.
IPSec Session Encryption Key	Plaintext in RAM	VPN traffic encryption/decryption using AES (128-,256-bit)	Overwritten with random pattern when no longer needed.
IKE SKEYSEED	Plaintext in RAM	Used to generate IKE protocol keys	Overwritten with random pattern when no longer needed.
IKE Pre-Shared Key	AES encrypted in Flash	Used to generate IKE protocol keys	Overwritten with random pattern when no longer needed
IKE Authentication Key	Plaintext in RAM	IKE peer-to-peer authentication using HMAC	Overwritten with random pattern when no longer needed.
IKE Key Generation Key	Plaintext in RAM	IPsec SA keying material	Overwritten with random pattern 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 random pattern 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 random pattern 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 random pattern when no longer needed.
Diffie-Hellman Keys	Plaintext in RAM	Key agreement and key establishment	Overwritten with v when no longer needed.
EC Diffie-Hellman Keys	Plaintext in RAM	Key agreement and key establishment	Overwritten with random pattern when no longer needed.
Firmware Update Key	Plaintext in RAM	Verification of firmware integrity when updating to	Overwritten with random pattern 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 random pattern 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 random pattern 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 random pattern 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 random pattern 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 random pattern when no longer needed.
SSH Session Encryption Key	Plaintext in RAM	AES (128-, 256-bit) key used for SSH session encryption	Overwritten with random pattern when no longer needed.
Locally Stored Passwords	SHA-1 hash in Flash	User authentication	Overwritten with random pattern when no longer needed.
Configuration Encryption Key	Plaintext in Flash	AES 256-bit key used to encrypt CSPs on the Boot device	Overwritten with random pattern when no longer needed.

6.2.3 800-56B conformance statements

32 The TOE fulfils the NIST SP 800-56B requirements listed below without extensions. The TOE does not implement any functionality within the SP 800-56B standard that is listed as “should not” and “shall not”.

33 Specifically, the TOE claims conformance to:

a) Section 5.9 (Key Derivation Functions for Key Establishment Schemes);

b) Section 6.3.1 (RSAKPG1 Family: rsakpg1-basic RSA Key Pair Generation with a Fixed Public Exponent);

c) Section 6.3.2 (RSAKPG2 Family: rsakpg1-basic RSA Key Pair Generation with a Random PublicExponent);

d) Section 6.4 (Assurances of Validity);

e) Section 6.4.1 (Assurance of Key Pair Validity);

f) Section 6.4.2 (Recipient Assurances of Public Key Validity);

g) Section 8 (Key Agreement Schemes); and

h) Section 9 (IFC based Key Transport Schemes).

6.2.4 Entropy and DRBG

34 The TOE makes use of two distinct entropy sources:

a) **Token.** Wide-band radio frequency (RF) white noise source provided by the Fortinet Entropy Token. Used for software-based cryptographic operations.

b) **CP9.** Oscillator based entropy source. Used for hardware accelerated cryptographic operations.

35 Entropy from the noise source is extracted, conditioned and used to seed the CTR_DRBG in use with 256 bits of full entropy. Additional detail regarding these entropy sources is provided the proprietary Entropy Description.

6.3 HTTPS/TLS

SFRs:	FCS_HTTPS_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_RSA_WITH_AES_128_CBC_SHA

b) TLS_RSA_WITH_AES_256_CBC_SHA

c) TLS_DHE_RSA_WITH_AES_128_CBC_SHA

d) TLS_DHE_RSA_WITH_AES_256_CBC_SHA

e) TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

f) TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

g) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

h) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

i) TLS_RSA_WITH_AES_128_CBC_SHA256

j) TLS_RSA_WITH_AES_256_CBC_ SHA256
k) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
l) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
m) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
n) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
o) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
p) 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 RSA, DHE, and ECDHE key agreement schemes. The RSA key agreement parameters are restricted to 2048 bits and are hardcoded into the server. 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.

6.3.3 TLS Client

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

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

a) TLS_RSA_WITH_AES_128_CBC_SHA
b) TLS_RSA_WITH_AES_256_CBC_SHA
c) TLS_DHE_RSA_WITH_AES_128_CBC_SHA
d) TLS_DHE_RSA_WITH_AES_256_CBC_SHA
e) TLS_DHE_RSA_WITH_AES_128_CBC_SHA256
f) TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
g) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
h) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
i) TLS_RSA_WITH_AES_128_CBC_SHA256
j) TLS_RSA_WITH_AES_256_CBC_ SHA256
k) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
l) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA
m) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
n) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
o) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
p) TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

44 Ciphersuites are not user-configurable.

45 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).

46 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.

47 The TLS client does not support certificate pinning.

48 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.

49 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

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

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

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

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

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

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

56 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).

6.5 IPsec

SFRs:	FCS_IPSEC_EXT.1

57 The TOE implements IPsec in accordance with RFC 4301.

58 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.

59 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).

60 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.

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

62 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.

63 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).

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

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

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

67 The TOE permits the configuration of IKEv1 Phase 2 SA and IKEv2 Child SA lifetimes in seconds, between 120 and 172800 (48 hours). IKEv2 Child SA lifetimes may also be configured based on number of bytes.

68 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.

69 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.

70 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.

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

72 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).

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

74 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.

75 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

76 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

77 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.

78 The TOE enforces a password policy. Administrative passwords may be at least 15 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.

79 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

80 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).

81 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;

82 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.

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

84 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.

85 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.

86 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

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

88 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.

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

90 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_SMR.2

91 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.

92 The TOE defines a single role, which is that of the Security Administrator. 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) Modify, delete, generate and/or import cryptographic keys;

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.

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

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

94 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.

95 The TOE performs the following self-tests upon initialisation:

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 will compare the image of the TOE to a known checksum of the image prior to booting.

c) Noise source tests

   i) The noise source(s) are 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

96 The above tests ensure that the CPU and memory utilised 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.

97 The cryptographic functionality will not be available if the cryptopraphic 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.

98 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.

99 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.

100 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.

101 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

102 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.

103 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

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

105 The TOE permits administrators to define a session lifetime for both local and remote sessions. Once this time limit has been met, the TOE will automatically close the active session (local or remote) and require TOE administrators to re-authenticate before any access to TSF data is permitted. TOE administrators may also manually close their sessions.

106 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_TRP.1/Admin

107 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).

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

109 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.

110 These protocols implement cryptographic algorithms to provide data transport security and integrity, preventing unauthorised 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

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

a) ICMPv4 (RFC 792)
   i) Type; and
   ii) Code

b) ICMPv6 (RFC 4443)
   i) Type; and
   ii) Code

c) IPv4 (RFC 791)
   i) Source address;
   ii) Destination Address; and
   iii) Transport Layer Protocol

d) IPv6 (RFC 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

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

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

114 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.

115 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.

116 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.

117 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.

118 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.

119 The TOE utilises 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 utilised in the management of sessions.

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

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

122 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).

123 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.

124 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.

125 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.

126 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).

127 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.

128 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.

129 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

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

131 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.

132 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.

133 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.

134 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).

135 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.

136 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.

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

138 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.

139 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.

140 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.

141 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.

142 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);`

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

144 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.

145 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.

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

147 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 or denying the traffic flow.

148 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.

149 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.

150 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).

151 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.

152 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

153 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

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

7.3 Security Requirements Rationale

155 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

156 This annex reproduces extended components definition from the claimed Protection Profiles where available.

157 The following extended components are taken from the VPN_EP which does not provide a an extended components definition:

a) FCS_IPSEC_EXT.1 (augments the base FWcPP version)

b) FIA_PSK_EXT.1

c) FPF_RUL_EXT.1

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

a) IPS_ABD_EXT.1

b) IPS_IPB_EXT.1

c) IPS_NTA_EXT.1

d) IPS_SBD_EXT.1

FWcPP Extended Components

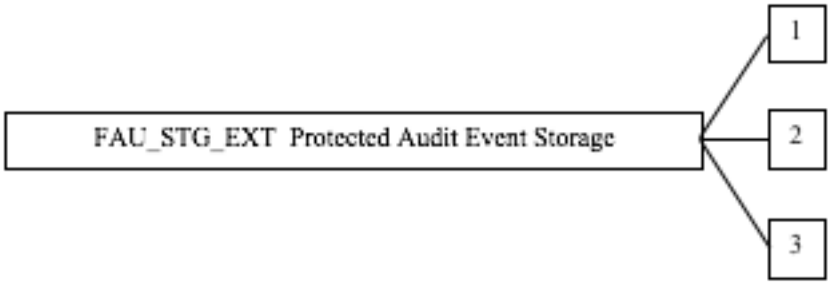
C.1 Security Audit (FAU)

C.1.1 Protected audit event storage (FAU_STG_EXT)

Family Behaviour

159 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 leveling



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

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

162 FAU_STG_EXT.3 Display warning for local storage space requires the TSF to generate a warning before the audit log becomes full.

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

163 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

164 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_ 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.

Application Note 106

For selecting the option of transmission of generated audit data to an external IT entity the TOE relies on a non-TOE audit server for storage and review of audit records. The storage of these audit records and the ability to allow the administrator to review these audit records is provided by the operational environment in that case.

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

FAU_STG_EXT.1.3 The TSF shall [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.

Application Note 107

The external log server might be used as alternative storage space in case the local storage space is full. The 'other action' could in this case be defined as 'send the new audit data to an external IT entity'.

C.1.1.2 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.

Application Note 108

This option should be chosen if the TOE supports this functionality.

In case the local storage for audit records is cleared by the administrator, the counters associated with the selection in the SFR should be reset to their initial value (most likely to 0). The guidance documentation should contain a warning for the administrator about the loss of audit data when he clears the local storage for audit records.

C.1.1.3 FAU_ STG_EXT.3 Display warning for local storage space

FAU_STG_EXT.3 Display warning for local storage space

FAU_STG_EXT.3.1 The TSF shall generate a warning to inform the user before the local space to store audit data is used up and/or the TOE will lose audit data due to insufficient local space.

Application Note 109

This option should be chosen if the TOE generates as warning to inform the user before the local storage space for audit data is used up. This might be useful if auditable events are stored on local storage space only.

It has to be ensured that the warning message required by FAU_STG_EXT.1.3 can be communicated to the user. The communication should be done via the audit log itself because it cannot be guaranteed that an administrative session is active at the time the event occurs.

C.2 Cryptographic Support (FCS)

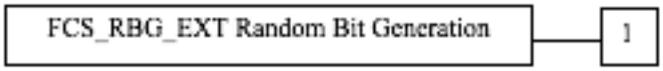
C.2.1 Random Bit Generation (FCS_RBG_EXT)

C.2.1.1 FCS_RBG_EXT.1 Random Bit Generation

Family Behaviour

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

Component leveling



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

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

a) There are no management activities foreseen

Audit: FCS_RBG_EXT.1

167 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 hardware-based sources] hardware-based noise source] with 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.

Application Note 110

For the first selection in FCS_RBG_EXT.1.2, the ST selects at least one of the types of noise sources. If the TOE contains multiple noise sources of the same type, the ST author fills the assignment with the appropriate number for each type of source (e.g., 2 software-based noise sources, 1 hardware-based noise source). The documentation and tests required in the Evaluation Activity for this element necessarily describes each source indicated in the ST.

ISO/IEC 18031:2011 contains three different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used, and include the specific underlying cryptographic primitives used in the requirement. While any of the identified hash functions (SHA-1, SHA-224, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES-based implementations for CTR_DRBG are allowed.

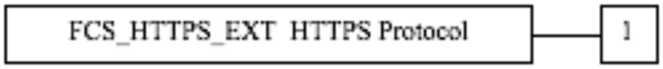
C.2.2 Cryptographic Protocols (Extended – FCS_HTTPS_EXT, FCS_ IPSEC_EXT, FCS_SSHC_EXT, FCS_SSHS_EXT, FCS_TLSC_EXT, FCS_TLSS_EXT)

C.2.2.1 FCS_HTTPS_EXT.1 HTTPS Protocol

Family Behaviour

168 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 leveling



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

Management: FCS_HTTPS_EXT.1

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

a) There are no management activities foreseen.

Audit: FCS_HTTPS_EXT.1

171 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_TLS_EXT.1 TLS 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.

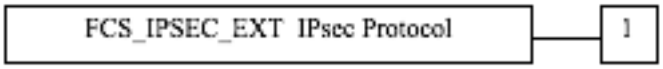
FCS_HTTPS_EXT.1.3 The TSF shall [selection: not establish the connection, request authorization to establish the connection, [assignment: other action]] if the peer certificate is deemed invalid.

C.2.2.2 FCS_IPSEC_EXT.1 IPsec Protocol

Family Behaviour

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

Component leveling



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

Management: FCS_IPSEC_EXT.1

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

a) Maintenance of SA lifetime configuration

Audit: FCS_IPSEC_EXT.1

174 The following actions should be auditable 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 Verification)

FCS_COP.1/Hash Cryptographic Operation (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.

Application Note 111

RFC 4301 calls for an IPsec implementation to protect IP traffic through the use of a Security Policy Database (SPD). The SPD is used to define how IP packets are to be handled: PROTECT the packet (e.g., encrypt the packet), BYPASS the IPsec services (e.g., no encryption), or DISCARD the packet (e.g., drop the packet). The SPD can be implemented in various ways, including router access control lists, firewall rulesets, a "traditional" SPD, etc. Regardless of the implementation details, there is a notion of a "rule" that a packet is "matched" against and a resulting action that takes place.

While there must be a means to order the rules, a general approach to ordering is not mandated, as long as the SPD can distinguish the IP packets and apply the rules accordingly. There may be multiple SPDs (one for each network interface), but this is not required.

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 and [selection: tunnel mode, no other 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, AES-CBC-256 (both specified by RFC 3602) and [selection: AES-GCM-128 (specified in RFC 4106), AES-GCM-256 (specified in RFC 4106), no other algorithms] together with a Secure Hash Algorithm (SHA)-based HMAC.

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 AES-CBC-128, AES-CBC-256 as specified in RFC 3602 and [selection: AES-GCM-128, AES-GCM-256 as specified in RFC 5282, no other algorithm].

Application Note 112

AES-GCM-128 and AES-GCM-256 may only be selected if IKEv2 is also selected, as there is no RFC defining AES-GCM for IKEv1.

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 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 configured within [assignment: integer range including 24] hours

]

].

Application Note 113

The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either volume-based lifetimes or time-based lifetimes (or a combination). This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE). Hardcoded limits do not meet this requirement. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the guidance documentation generated for AGD_OPE.

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;

]

].

Application Note 114

The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either volume-based lifetimes or time-based lifetimes (or a combination). This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE). Hardcoded limits do not meet this requirement. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the guidance documentation generated for AGD_OPE.

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange (“x” in gx 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.

Application Note 115

For DH groups 19 and 20, the "x" value is the point multiplier for the generator point G.

Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1.9 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment for this element. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.

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

- [assignment: 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

].

Application Note 116

The ST author must select the second option for nonce lengths if IKEv2 is also selected (as this is mandated in RFC 5996). The ST author may select either option for IKEv1.

For the first option for nonce lengths, since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1.10 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment for this element. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.

Because nonces may be exchanged before the DH group is negotiated, the nonce used should be large enough to support all TOE-chosen proposals in the exchange.

FCS_IPSEC_EXT.1.11 The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), and [selection: 19 (256-bit Random ECP), 5 (1536-bit MODP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP), [assignment: other DH groups that are implemented by the TOE], no other DH groups].

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.

Application Note 117

The ST author chooses either or both of the IKE selections based on what is implemented by the TOE. While it is acceptable for this capability to be configurable, the default configuration in the evaluated configuration *(either "out of the box" or by configuration guidance in the AGD documentation) must enable this functionality.*

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].

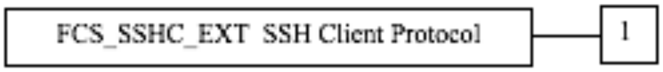
FCS_IPSEC_EXT.1.14 The TSF shall only establish a trusted channel to peers with valid certificates.

C.2.2.3 FCS_SSHC_EXT.1 SSH Client

Family Behaviour

175 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 leveling



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

Management: FCS_SSHC_EXT.1

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

a) There are no management activities foreseen.

Audit: FCS_SSHC_EXT.1

178 The following actions should be auditable 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_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Verification)
FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

FCS_SSHC_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [selection: 5647, 5656, 6187, 6668, no other RFCs].

Application Note 118

The ST author selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the assurance activity for this requirement.

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, password-based.

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.

Application Note 119

RFC 4253 provides for the acceptance of "large packets" with the caveat that the packets should be of "reasonable length" or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining "reasonable length" for the TOE.

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 transport implementation uses [assignment: List of public key algorithms] 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 the SSH connection be rekeyed after no more than 2^28 packets have been transmitted using that key.

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 or [selection: a list of trusted certification authorities, no other methods] as described in RFC 4251 section 4.1.

Application Note 120

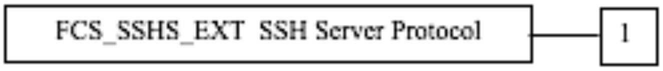
The list of trusted certification authorities can only be selected if x509v3-ecdsa-sha2-nistp256 or x509v3-ecdsa-sha2-nistp384 are specified in FCS_SSHC_EXT.1.5.

C.2.2.4 FCS_SSHS_EXT.1 SSH Server Protocol

Family Behaviour

179 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 leveling



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

Management: FCS_SSHS_EXT.1

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

a) There are no management activities foreseen.

Audit: FCS_SSHS_EXT.1

182 The following actions should be auditable 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_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Verification)
FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [selection: 5647, 5656, 6187, 6668, no other RFCs].

Application Note 121

The ST author selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the assurance activity for this requirement.

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 [assignment: number of bytes] bytes in an SSH transport connection are dropped.

Application Note 122

RFC 4253 provides for the acceptance of “large packets” with the caveat that the packets should be of “reasonable length” or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining “reasonable length” for the TOE.

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 transport implementation uses [assignment: List of public key algorithms] 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.

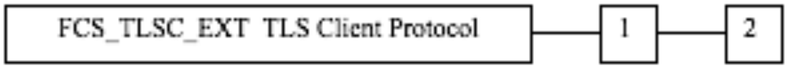
FCS_SSHS_EXT.1.8 The TSF shall ensure that the SSH connection be rekeyed after no more than 2^28 packets have been transmitted using that key.

C.2.2.5 FCS_TLSC_EXT TLS Client Protocol

Family Behaviour

183 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 leveling



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

185 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

186 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

187 The following actions should be auditable 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

Hierarchical to: No other components

Dependencies: FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Verification)
FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSC_EXT.1.1 The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

- Mandatory Ciphersuites:
  - [assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]
- [selection: Optional Ciphersuites:
  - [assignment: List of optional ciphersuites and reference to RFC in which each is defined]
  - no other ciphersuite]].

Application Note 123

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS_RSA_WITH_AES_128_CBC_SHA is required in order to ensure compliance with RFC 5246.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

Application Note 124

The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

FCS_TLSC_EXT.1.3 The TSF shall only establish a trusted channel if the peer certificate is valid.

Application Note 125

Validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

FCS_TLSC_EXT.1.4 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [assignment: List of supported curves including an option for 'none'].

Application Note 126

If ciphersuites with elliptic curves were selected in FCS_TLSC_EXT.1.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS_TLS_EXT.1.1, then 'none' should be selected.

This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1/SigGen and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.

FCS_TLSC_EXT.2 TLS Client Protocol with Authentication

Hierarchical to: FCS_TLSC_EXT.1 TLS Client Protocol

Dependencies: FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Verification)
FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSC_EXT.2.1 The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

- Mandatory Ciphersuites:
  - [assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]
- [selection: Optional Ciphersuites:
  - [assignment: List of optional ciphersuites and reference to RFC in which each is defined]
  - no other ciphersuite]].

Application Note 127

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS_RSA_WITH_AES_128_CBC_SHA is required in order to ensure compliance with RFC 5246.

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

Application Note 128

The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

FCS_TLSC_EXT.2.3 The TSF shall only establish a trusted channel if the peer certificate is valid.

Application Note 129

Validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

FCS_TLSC_EXT.2.4 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [assignment: List of supported curves including an option for 'none'].

FCS_TLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

Application Note 130

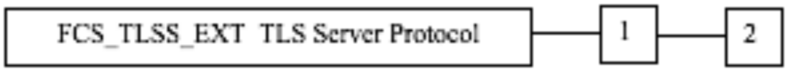
The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.

C.2.2.6 FCS_TLSS_EXT TLS Server Protocol

Family Behaviour

188 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 leveling



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

190 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

191 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

192 The following actions should be auditable 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

Hierarchical to: No other components

Dependencies: FCS_CKM.1 Cryptographic Key Generation
FCS_COP.1/DataEncryption Cryptographic operation (AES Data

encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Verification)
FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSS_EXT.1.1 The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

- Mandatory Ciphersuites:
    - [assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]
- [selection: Optional Ciphersuites:
    - [assignment: List of optional ciphersuites and reference to RFC in which each is defined]
    - no other ciphersuite]].

Application Note 131

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS_RSA_WITH_AES_128_CBC_SHA is required in order to ensure compliance with RFC 5246.

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

Application Note 132

Any TLS versions not selected in FCS_TLSS_EXT.1.1 should be selected here.

FCS_TLSS_EXT.1.3 The TSF shall generate key establishment parameters using RSA with key size 2048 bits and [selection: 3072 bits, 4096 bits, no other size] and [selection: [assignment: List of elliptic curves]; [assignment: List of diffie-hellman parameter sizes]].

Application Note 133

The assignments will be filled in based on the assignments performed in FCS_TLSS_EXT.1.1.

FCS_TLSS_EXT.2 TLS Server Protocol with mutual authentication

Hierarchical to: FCS_TLSS_EXT.1 TLS Server Protocol

Dependencies: FCS_CKM.1 Cryptographic Key Generation
FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Verification)
FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSS_EXT.2.1 The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

- Mandatory Ciphersuites:

  - [assignment: List of mandatory ciphersuites and reference to RFC in which each is defined]
- [selection: Optional Ciphersuites:
  - [assignment: List of optional ciphersuites and reference to RFC in which each is defined]
  - no other ciphersuite]].

Application Note 134

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS_RSA_WITH_AES_128_CBC_SHA is required in order to ensure compliance with RFC 5246.

FCS_TLSS_EXT.2.2 The TSF shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, and [selection: TLS 1.1, TLS 1.2, none].

Application Note 135

Any TLS versions not selected in FCS_TLSS_EXT.2.1 should be selected here.

FCS_TLSS_EXT.2.3 The TSF shall generate key establishment parameters using RSA with key size 2048 bits and [selection: 3072 bits, 4096 bits, no other size] and [selection: [assignment: List of elliptic curves]; [assignment: List of diffie-hellman parameter sizes]].

Application Note 136

The assignments will be filled in based on the assignments performed in FCS_TLSS_EXT.2.1.

FCS_TLSS_EXT.2.4 The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

Application Note 137

The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include support for client-side certificates for TLS mutual authentication.

FCS_TLSS_EXT.2.5 The TSF shall not establish a trusted channel if the peer certificate is invalid.

Application Note 138

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

FCS_TLSS_EXT.2.6 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 peer.

Application Note 139

This requirement only applies to those TOEs performing mutually-authenticated TLS (FCS_TLSS_EXT.2.4). The peer identifier may be in the Subject field or the Subject Alternative Name extension of the certificate. The expected identifier may either be configured, may be compared to the Domain Name, IP address, username, or email address used by the peer, or may be passed to a directory server for comparison.

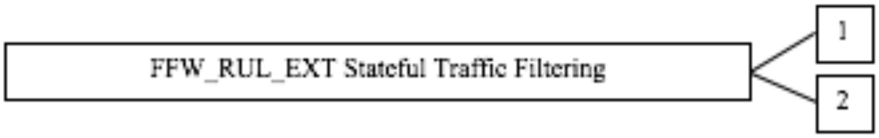
C.3 Firewall (FFW)

C.3.1 Stateful Traffic Filter Firewall (FFW_RUL_EXT)

Family Behaviour

193 This requirement is used to specify the behavior of a Stateful Traffic Filter Firewall. The network protocols that the TOE can filter, as well as the attributes that can be used by an administrator to construct a ruleset are identified in this component. How the ruleset is processed (i.e., ordering) is specified, as well as any expected default behavior on the part of the TOE.

Component leveling



194 FFW_RUL_EXT.1 Stateful Traffic Filtering requires the TOE to filter network traffic based on a ruleset configured by an authorized administrator.

Management: FFW_RUL_EXT.1

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

a) enable/disable a ruleset on a network interface

b) configure a ruleset

c) specifying rules that govern the use of resources

Audit: FFW_RUL_EXT.1

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

a) Minimal:

- Result (i.e., drop, allow) of applying a rule in the ruleset to a network packet
- Configuration of the ruleset
- Indication of packets dropped due to too much network traffic

C.3.1.1 FFW_RUL_EXT.1 Stateful Traffic Filtering

FFW_RUL_EXT.1 Stateful Traffic Filtering

Hierarchical to: No other components

Dependencies: None

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: [assignment: list of attributes supported by the ruleset].

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: [assignment: list of supported protocols for which state is maintained] based on the following network packet attributes: [assignment: list of attributes associated with each of the protocols].

b) Remove existing traffic flows from the set of established traffic flows based on the following: [selection: 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: [assignment: list of default rules that are applied to network traffic flow].

FFW_RUL_EXT.1.7 The TSF shall be capable of dropping and logging according to the following rules: [assignment: list of specific rules that the TOE is capable of enforcing]

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 configured number of [assignment: rules governing the use of resources].

C.3.1.2 FFW_RUL_EXT.2 Stateful Filtering of Dynamic Protocols

FFW_RUL_EXT.2 Stateful Filtering of Dynamic Protocols

Hierarchical to: No other components

Dependencies: None

FFW_RUL_EXT.2.1 The TSF shall dynamically define rules or establish sessions allowing network traffic to flow for the following network protocols [assignment: list of supported protocols].

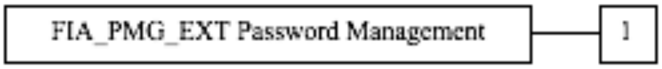
C.4 Identification and Authentication (FIA)

C.4.1 Password Management (FIA_PMG_EXT)

Family Behaviour

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

Component leveling



198 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

199 No management functions.

Audit: FIA_PMG_EXT.1

200 No specific audit requirements.

C.4.1.1 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]];

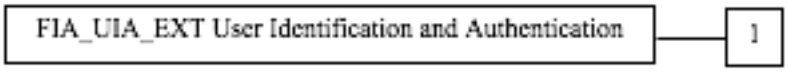
b) Minimum password length shall be settable by the Security Administrator, and support passwords of 15 characters or greater.

C.4.2 User Identification and Authentication (FIA_UIA_EXT)

Family Behaviour

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

Component leveling



202 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

203 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

204 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.4.2.1 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, [assignment: list of services, actions performed by the TSF in response to non-TOE requests.]]

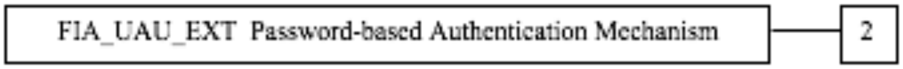
FIA_UIA_EXT.1.2 The TSF shall require each administrative user to be successfully identified and authenticated before allowing any other TSF-mediated actions on behalf of that administrative user.

C.4.3 User authentication (FIA_UAU) (FIA_UAU_EXT)

Family Behaviour

205 Provides for a locally based administrative user authentication mechanism

Component leveling



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

Management: FIA_UAU_EXT.2

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

a) None

Audit: FIA_UAU_EXT.2

208 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.4.3.1 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2 Password-based Authentication Mechanism

Hierarchical to: No other components.

Dependencies: None

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [selection: [assignment: other authentication mechanism(s)], none] to perform administrative user authentication.

C.4.4 Authentication using X.509 certificates (Extended – FIA_X509_EXT)

Family Behaviour

209 This family defines the behavior, 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 leveling

FIA_X509_EXT X509 Certificate
1
2
3

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

211 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.

212 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/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3

213 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/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3

214 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.4.4.1 FIA_X509_EXT.1/Rev X.509 Certificate Validation

FIA_X509_EXT.1/Rev X.509 Certificate Validation

Hierarchical to: No other components

Dependencies: No other components

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.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [selection: the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759].
- 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].

Application Note 140

FIA_X509_EXT.1.1 lists the rules for validating certificates. The ST author selects whether revocation status is verified using OCSP or CRLs. The ST author fills in the assignment with rules that may apply to other requirements in the ST. For instance, if a protocol such as TLS that uses certificates is specified in the ST, then certain values for the extendedKeyUsage field (e.g., “Server Authentication Purpose”) could be specified.

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.

Application Note 141

This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.

C.4.4.2 FIA_X509_EXT.2 X509 Certificate Authentication

FIA_X509_EXT.2 X.509 Certificate Authentication

Hierarchical to: No other components

Dependencies: No other components

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

Application Note 142

If the TOE specifies the implementation of communications protocols that perform peer authentication using certificates, the ST author either selects or assigns the protocols that are specified; otherwise, they select "no protocols". The TOE may also use certificates for other purposes; the second selection and assignment are used to specify these cases.

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].

Application Note 143

Often a connection must be established to check the revocation status of a certificate - either to download a CRL or to perform a lookup using OCSP. The selection is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate valid according to all other rules in FIA_X509_EXT.1/Rev, the behavior indicated in the selection determines the validity.

C.4.4.3 FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3 X.509 Certificate Requests

Hierarchical to: No other components

Dependencies: No other components

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 [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.

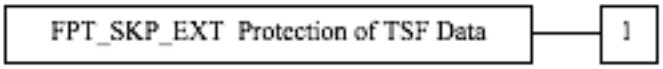
C.5 Protection of the TSF (FPT)

C.5.1 Protection of TSF Data (FPT_SKP_EXT)

Family Behaviour

215 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 leveling



216 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

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

a) There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

218 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.5.1.1 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.

Application Note 144

The intent of this requirement is for the device to protect keys, key material, and authentication credentials from unauthorized disclosure. This data should only be accessed for the purposes of their assigned security functionality, and there is no need for them to be displayed/accessed at any other time. This requirement does not prevent the device from providing indication that these exist, are in use, or are still valid. It does, however, restrict the reading of the values outright.

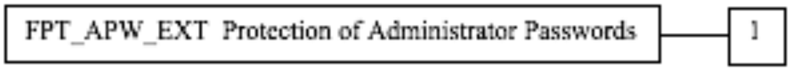
C.5.2 Protection of Administrator Passwords (FPT_APW_EXT)

C.5.2.1 FPT_APW_EXT.1 Protection of Administrator Passwords

Family Behaviour

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

Component leveling



220 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

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

a) No management functions.

Audit: FPT_APW_EXT.1

222 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 passwords in non-plaintext form.

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

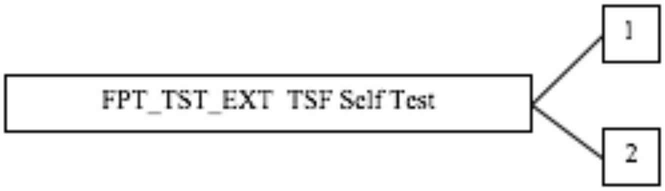
C.5.3 TSF self test

C.5.3.1 FPT_TST_EXT.1 TSF Testing

Family Behaviour

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

Component leveling



224 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.

225 FPT_TST_EXT.2 Self tests based on certificates applies when using certificates as part of self test, and requires that the self test fails if a certificate is invalid.

Management: FPT_TST_EXT.1, FPT_TST_EXT.2

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

a) No management functions.

Audit: FPT_TST_EXT.1, FPT_TST_EXT.2

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

a) Indication that TSF self test was completed

FPT_TST_EXT.1 TSF testing

Hierarchical to: No other components.

Dependencies: None

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].

Application Note 145

It is expected that self-tests are carried out during initial start-up (on power on). Other options should only be used if the developer can justify why they are not carried out during initial start-up. It is expected that at least self-tests for verification of the integrity of the firmware and software as well as for the correct operation of cryptographic functions necessary to fulfil the SFRs will be performed. If not all self-test are performed during start-up multiple iterations of this SFR are used with the appropriate options selected. In future versions of this cPP the suite of self-tests will be required to contain at least mechanisms for measured boot including self-tests of the components which perform the measurement.

Application Note 146

If certificates are used by the self-test mechanism (e.g. for verification of signatures for integrity verification), certificates are validated in accordance with FIA_X509_EXT.1 and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TST_EXT.2 must be included in the ST.

FPT_TST_EXT.2 Self tests based on certificates

Hierarchical to: No other components.

Dependencies: None

FPT_TST_EXT.2.1 The TSF shall fail self-testing if a certificate is used for self tests and the corresponding certificate is deemed invalid.

Application Note 147

Certificates may optionally be used for self-tests (FPT_TST_EXT.1.1). This element must be included in the ST if certificates are used for self-tests. If “code signing for integrity verification” is selected in FIA_X509_EXT.2.1, FPT_TST_EXT.2 must be included in the ST.

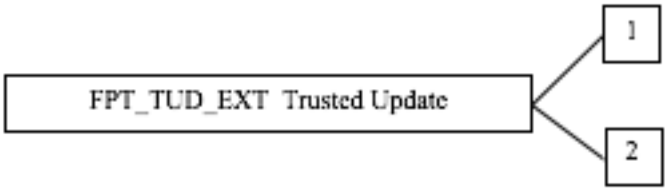
Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with FIA_X509_EXT.1.

C.5.4 Trusted Update (FPT_TUD_EXT)

Family Behaviour

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

Component leveling



229 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.

230 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

231 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

232 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.5.4.1 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1 Trusted update

Hierarchical to: No other components

Dependencies: FCS_COP.1/DataEncryption Cryptographic operation (for cryptographic signature), or FCS_COP.1/Hash Cryptographic operation (for cryptographic hashing)

FPT_TUD_EXT.1.1 The TSF shall provide [assignment: authorised users] the ability to query the currently executing version of the TOE firmware/software as well as the most recently installed version of the TOE firmware/software.

Application Note 148

The version currently running (being executed) may not be the version most recently installed. For instance, maybe the update was installed but the system requires a reboot before this update will run. Therefore, it needs to be clear that the query should indicate both the most recently executed version as well as the most recently installed update.

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

Application Note 149

The selection in FPT_TUD_EXT.1.2 distinguishes the support of automatic checking for updates and support of automatic updates. The first option refers to a TOE that checks whether a new update is available, communicates this to the administrator (e.g. through a message during an administrator session, through log files) but requires some action by the administrator to actually perform the update. The second option refers to a TOE that checks for updates and automatically installs them upon availability.

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

Application Note 150

The digital signature mechanism referenced in the selection of FPT_TUD_EXT.1.3 is one of the algorithms specified in FCS_COP.1/SigGen.The published hash referenced in FPT_TUD_EXT.1.3 is generated by one of the functions specified in FCS_COP.1/Hash. The ST author should choose the mechanism implemented by the TOE; it is acceptable to implement both mechanisms.

Application Note 151

Future versions of this cPP will mandate the use of a digital signature mechanism for trusted updates.

Application Note 152

If certificates are used by the update verification mechanism, certificates are validated in accordance with FIA_X509_EXT.1 and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TUD_EXT.2 must be included in the ST.

Application Note 153

“Update” in the context of this SFR refers to the process of replacing a non-volatile, system resident software component with another. The former is referred to as the NV image, and the latter is the update image. While the update image is typically newer than the NV image, this is not a requirement. There are legitimate cases where the system owner may want to rollback a component to an older version (e.g. when the component manufacturer releases a faulty update, or when the system relies on an undocumented feature no longer present in the update). Likewise, the owner may want to update with the same version as the NV image to recover from faulty storage.

All discrete software components (e.g. applications, drivers, kernel, firmware) of the TSF, should be digitally signed by the corresponding manufacturer and subsequently verified by the mechanism performing the *update. Since it is recognized that components may be signed by different manufacturers, it is essential that the update process verify that both the update and NV images were produced by the same manufacturer (e.g. by comparing public keys) or signed by legitimate signing keys (e.g. successful verification of certificates when using X.509 certificates).*

C.5.4.2 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 not install an update if the code signing certificate is deemed invalid.

FPT_TUD_EXT.2.2 When the certificate is deemed invalid because the certificate has expired, the TSF shall [selection: allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate].

Application Note 154

Certificates may optionally be used for code signing of system software updates (FPT_TUD_EXT.1.3). This element must be included in the ST if certificates are used for validating updates. If “code signing for system software updates” is selected in FIA_X509_EXT.2.1, FPT_TUD_EXT.2 must be included in the ST.

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with FIA_X509_EXT.1. For expired certificates the author of the ST selects whether the certificate shall be accepted, rejected or the choice is left to the administrator to accept or reject the certificate.

C.6 TOE Access (FTA)

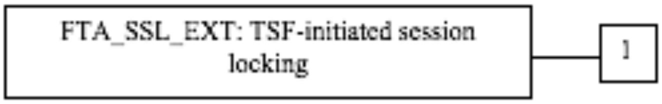
C.6.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

Family Behaviour

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

234 The extended FTA_SSL_EXT family is based on the FTA_SSL family.

Component leveling



235 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

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

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

Audit: FTA_SSL_EXT.1

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

a) Any attempts at unlocking an interactive session.

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 user'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.

Annex B: CAVP Certificates

Annex B.1: SFR Coverage

Table 20: CAVP SFR Coverage Mapping

SFR	Selections	Usage	CAVP	Notes
FCS_CKM.1 Cryptographic Key Generation	RSA	TLS, SSH	C644	
	ECC	TLS	C644	
	FFC – DH Group 14	IPsec, TLS, SSH	n/a	
FCS_CKM.1/IKE Cryptographic Key Generation (for IKE Peer Authentication)	RSA	IPsec	C644	Certificate / key gen performed by SSL library.
	ECDSA	IPsec	C644	Certificate / key gen performed by SSL library.
			C531	Hardware acceleration per Table 21.
FCS_CKM.2 Cryptographic Key Establishment	RSA	TLS (Audit Server) TLS/HTTPS (GUI)	n/a	
	ECC	TLS (Audit Server) TLS/HTTPS (GUI)	C644	
		IPsec (VPN)	C599	
		IPsec (VPN) TLS (Audit Server) TLS/HTTPS (GUI)	C531	Hardware acceleration per Table 21.

SFR	Selections	Usage	CAVP	Notes
	DH Group 14	IPsec (VPN) TLS (Audit Server) TLS/HTTPS (GUI) SSH (Admin CLI)	n/a	
FCS_COP.1/DataEncryption	AES-CBC-128/256	TLS, SSH	C644	
			C531	Hardware acceleration per Table 21.
		IPsec	C599	
			C531	Hardware acceleration per Table 21.
	AES-GCM-128/256	TLS, SSH	C644	
			C531	Hardware acceleration per Table 21.
		IPsec	C599	
			C531	Hardware acceleration per Table 21.
FCS_COP.1/SigGen	RSA	IPsec, TLS, SSH, Trusted Update	C644	
			C531	Hardware acceleration per Table 21.
	ECDSA	TLS	C644	
			C531	Hardware acceleration per Table 21.

SFR	Selections	Usage	CAVP	Notes
		IPsec	C599	
			C531	Hardware acceleration per Table 21.
FCS_COP.1/Hash	SHA-1, SHA-256, SHA-384, SHA-512	IPsec, Password Hashing	C599	
		TLS, SSH	C644	
		IPsec, TLS, SSH	C531	Hardware acceleration per Table 21.
FCS_COP.1/KeyedHash	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	IPsec	C599	
		TLS, SSH	C644	
		IPsec, TLS, SSH	C531	Hardware acceleration per Table 21.
FCS_RBG_EXT.1	CTR_DRBG (AES)	TOE RBG	C613	

Annex B.2: CAVP Libraries

Table 21: CAVP Libraries & Capabilities Mapping

Library Name	CAVP	Capability	Usage
Fortinet FortiOS FIPS 6K series Cryptographic Library v5.6	C599	AES-CBC	Primarily IPsec
		AES-GCM	
		KDF IKEv1	

Library Name	CAVP	Capability	Usage
		KDF IKEv2	
		HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	
		SHA-1, SHA-256, SHA-384, SHA-512	
Fortinet FortiOS SSL 6K series Cryptographic Library v5.6	C644	AES-CBC	Primarily TLS and SSH
		AES-GCM	
		KDF SSH	
		KDF 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)	

Library Name	CAVP	Capability	Usage
		SHA-1, SHA-256, SHA-384, SHA-512	
Fortinet FortiOS RBG 6K series Cryptographic Library v5.6	C613	Counter DRBG	TOE DRBG
Fortinet CP9 Cryptographic Library v5.6	C531	AES-CBC	Hardware acceleration when configured (enabled by default).
		AES-GCM	
		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 SigGen (186-4) RSA SigVer (186-4)	
		SHA-1, SHA-256, SHA-384, SHA-512	