Cisco FTD 7.4 on Firepower 4100 and 9300 Series with FMC/FMCv

Security Target

ST Version 1.0

February 25, 2025

Table of Contents

List of Tables

List of Figures

List of Acronyms

The following acronyms and abbreviations are common and may be used in this Security Target:

Table 1: Acronyms

Acronyms/Abbreviations	Definition
ACL	Access Control List
AES	Advanced Encryption Standard
CC	Common Criteria
CEM	Common Evaluation Methodology
CM	Configuration Management
cpp_nd_v2.2e	Collaborative Protection Profile for Network Devices v2.2e
DHCP	Dynamic Host Configuration Protocol
EAL	Evaluation Assurance Level
EHWIC	Ethernet High-Speed WAN Interface Card
ESP	Encapsulating Security Payload
FMC	Firepower Management Center
FMCv	Firepower Management Center Virtual
FOM	FIPS Object Module
FTD	Firepower Threat Defense
Gbps	Gigabits per second
GE	Gigabit Ethernet port
HTTPS	Hyper-Text Transport Protocol Secure
ICMP	Internet Control Message Protocol
IKE	Internet Key Exchange
IPS	Intrusion Prevention System
IPsec	Internet Protocol Security
IT	Information Technology
mod_cpp_fw_v1.4e	PP-Module for Stateful Traffic Filter Firewalls v1.4e
mod_ips_v1.0	PP-Module for Intrusion Protection Systems (IPS) v1.0
mod_vpngw_v1.3	PP-Module for Virtual Private Network (VPN) Gateways v1.3
NGIPS	Cisco Next-Generation IPS
NDcPP	Network Device Collaborative Protection Profile
NGFW	Cisco Next-Generation Firewall
OE	Operational Environment
OS	Operating System
REST	Representational State Transfer
PoE	Power over Ethernet
POP3	Post Office Protocol
PP	Protection Profile
SA	Security Association
SFP	Small–form-factor pluggable port
SHA	Secure Hash Algorithm
SIP	Session Initiation Protocol
SSHv2	Secure Shell (version 2)
SSM	Security Services Module
SSP	Security Services Processor
ST	Security Target

Acronyms/Abbreviations	Definition
TCP	Transport Control Protocol
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Function
TSP	TOE Security Policy
UDP	User Datagram Protocol
VLAN	Virtual Local Area Network
VPN	Virtual Private Network
VS	Virtualization System
WAN	Wide Area Network
WIC	WAN Interface Card

DOCUMENT INTRODUCTION

Prepared By:

Cisco Systems, Inc.

170 West Tasman Dr.

San Jose, CA 95134

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), the Firepower Threat Defense (FTD) with Firepower Management Center (FMC). This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements, and the IT security functions provided by the TOE which meet the set of requirements. Administrators of the TOE will be referred to as administrators, authorized administrators, TOE administrators, semi-privileged, privileged administrators, and security administrators in this document.

1 SECURITY TARGET INTRODUCTION

The Security Target contains the following sections:

- Security Target Introduction [Section 1]
- Conformance Claims [Section 2]
- Security Problem Definition [Section 3]
- Security Objectives [Section 4]
- IT Security Requirements [Section 5]
- TOE Summary Specification [Section 6]
- Supplemental TOE Summary Specification Information [Section 7]
- References [Section 8]

The structure and content of this ST comply with the requirements specified in the Common Criteria (CC), Part 1, Annex A, and Part 2.

1.1 ST and TOE Reference

This section provides information needed to identify and control this ST and its TOE.

Table 2: ST and TOE Identification

Name	Description
ST Title	Cisco FTD 7.4 on Firepower 4100 and 9300 Series with FMC/FMCv Security Target
ST Version	1.0
Publication Date	February 25, 2025
Vendor and ST Author	Cisco Systems, Inc.
TOE Reference	Cisco FTD 7.4 on Firepower 4100 and 9300 Series with FMC and FMCv
TOE Hardware Models	• Firepower 4100 Series (4112, 4115, 4125 and 4145) • Firepower 9300 (including chassis, supervisor blade, and security module) • Cisco Firewall Management Center (FMC) (FMC1600, FMC2600, FMC4600, FMC1700, FMC2700 and FMC4700) • FMCv running on ESXi 7.0 on Unified Computing System (UCS) UCSC-C220-M5, UCSC-C240-M5, UCSC-C480-M5, UCSC-C220-M6, UCSC-C225-M6, UCSC-C240-M6, UCSC-C220-M7, UCSC-C240-M7 and UCS-E1100D-M6
TOE Software Version	FTD 7.4, FXOS 2.14 and FMC/FMCv 7.4
Keywords	Firewall, VPN Gateway, Router

1.2 TOE Overview

The Cisco Firepower 4100 and 9300 security appliances are purpose-built, scalable platforms with firewall, VPN and IPS capabilities provided by Firepower Threat Defense (FTD) software that is running on the Firepower eXtensible Operating System (FXOS). The TOE includes one or more Firepower appliances (running FTD and FXOS software) that are centrally managed by a Firepower Management Center (FMC) appliance, and together the FMC and Firepower (running FTD/FXOS) appliances form the TOE (Distributed TOE Use Case 3). The TOE includes the hardware models as defined in Table 2 of section 1.1.

1.2.1 TOE Product Type

Each appliance component of the TOE consists of hardware and software that provide connectivity and security services onto a single, secure device.

The models that comprise the TOE have common hardware characteristics (for example, the same FXOS image runs on all the models 4100 series and 9300, the same FTD image runs on the FXOS regardless of the platforms and the same FMC image runs on all the FMC appliances). These differing characteristics affect only non-TSF relevant functionality (such as throughput, processing speed, number and type of network connections supported, number of concurrent connections supported, and amount of storage) and therefore support security equivalency of the TOE in terms of hardware.

For firewall services, the FTD running on the security module provides application-aware stateful packet filtering firewalls. A stateful packet filtering firewall controls the flow of IP traffic by matching information contained in the headers of connection-oriented or connection-less IP packets against a set of rules specified by the authorized administrator for firewalls. This header information includes source and destination host (IP) addresses, source and destination port numbers, and the transport service application protocol (TSAP) held within the data field of the IP packet. Depending upon the rule and the results of the match, the firewall either passes or drops the packet. The stateful firewall remembers the state of the connection from information gleaned from prior packets flowing on the connection and uses it to regulate current packets. The packet will be denied if the security policy is violated.

In addition to IP header information, the TOE mediates information flows on the basis of other information, such as the direction (incoming or outgoing) of the packet on any given firewall network interface. For connection-oriented transport services, the firewall either permits connections and subsequent packets for the connection or denies the connection and subsequent packets associated with the connection.

The application-inspection capabilities automate the network to treat traffic according to detailed policies based not only on port, state, and addressing information, but also on application information buried deep within the packet header. By comparing this deep-packet inspection information with corporate policies, the firewall will allow or block certain traffic. For example, it will automatically drop application traffic attempting to gain entry to the network through an open port-even if it appears to be legitimate at the user and connection levels-if a business's corporate policy prohibits that application type from being on the network.

The TOE also provides IPsec connection capabilities. All references within this ST to “VPN” connectivity refer to the use of IPsec tunnels to secure connectivity to and/or from the TOE, for example, gateway-to-gateway[1] VPN or remote access VPN.

The TOE provides intrusion prevention system (IPS) capabilities by combining the security of a Next Generation IPS (NGIPS) with the power of access control, malware protection, and URL/IP filtering

[1] This is also known as site-to-site or peer-to-peer VPN.

known as Security Intelligence. The TOE monitors incoming and outgoing network traffic and performs real-time traffic analysis and logging using the Snort® engine. All packets on the monitored network are scanned, decoded, preprocessed and compared against a set of rules to determine whether inappropriate traffic, such as system attacks, is being sent over the network. The system generates alerts or blocks the traffic when deviations of the expected network behavior are detected or when there is a match to a known attack pattern.

1.2.2 Supported non-TOE Hardware/ Software/ Firmware

The TOE supports (in some cases optionally) the following hardware, software, and firmware in its environment when the TOE is configured in its evaluated configuration:

Table 3: IT Environment Components

Component	Required	Usage/Purpose Description for TOE performance
Management Workstation with SSH Client	Yes	This includes any IT Environment Management workstation with SSH client installed that is used by the TOE administrator to support TOE administration through SSHv2 protected channels. Any SSH client that supports SSHv2 may be used.
Management Workstation with Web Browser	Yes	This includes any IT Environment Management workstation with a web browser installed that is used by the TOE administrator to support TOE administration through TLS/HTTPS protected channels. Any browser that supports TLSv1.2 may be used.
Audit (syslog) Server	Yes	This includes any syslog server to which the TOE would transmit syslog messages. Connections to remote audit servers must be tunneled in IPsec or TLS.
Certification Authority	Yes	This includes any IT Environment Certification Authority on the TOE network. This can be used to provide the TOE with a valid certificate during certificate enrollment.
Remote Tunnel Endpoint	Yes	This includes any peer with which the TOE participates in tunneled communications. Remote tunnel endpoints may be any device or software client that supports IPsec tunneling. Both VPN clients and VPN gateways can be considered to be remote tunnel endpoints.
NTP Server	No	The TOE supports communications with an NTP server via IPsec tunnel.

1.3 TOE DESCRIPTION

This section provides an overview and description of the TOE. The TOE is comprised of both software and hardware. The models are comprised of the following: FP 4112, 4115, 4125, 4145 and 9300 and Firepower Management Center (FMC1600, FMC2600, FMC4600, FMC1700, FMC2700, FMC4700 and FMCv). The software is comprised of the FTD software image Release 7.4 (running directly on a 4100 series, or on a security module in a 9300), FXOS 2.14 (running on 4100 series or on the Supervisor blade of a 9300), and FMC (or FMCv) version 7.4.

The Cisco Firepower 9300 security appliance is a modular, scalable, carrier-grade appliance that includes the Chassis (including fans and power supply), Supervisor Blade[2] (to manage the security application running on the security module), network module (optional) and security module that contains the FTD software. The FP4100 Series appliance is a complete standalone, bundle unit that contains everything required above in one appliance. More details on the FP4100 is provided in sections 1.3 and 1.5.

Table 4: FP 9300 Components

Component	Required	Security-Relevant	Description
Chassis	Yes	No	Provides four fans to cool the entire system, two power supplies (AC or DC), and slots for the Supervisor blade, security module, and network module.
Supervisor Blade (or Module)	Yes	Yes	Running Firepower eXtensible Operating System (FXOS), this component is used to manage the FTD running on the security module. The processor for the Supervisor Blade is an Intel i3-3115C (Ivy Bridge).
Security Module	Yes	Yes	FP 9300 must have at least one security module (on which FTD is installed) in the evaluated configuration but can handle up to 3 security modules at a time. These are six types of security modules and the processors in each of them: • SM-40 (Intel Xeon Gold 6138 (Skylake) and CN3550 (NITROX III)) • SM-48 (Intel Xeon Platinum 8160 (Skylake) and CN5560 (NITROX V)) • SM-56 (Intel Xeon Platinum 8176 (Skylake) and CN5560 (NITROX V)) The throughputs for the listed security modules are 54, 64 and 70 Gbps respectively. Security module types cannot be mixed within a chassis.
Network Module	No	No	Provides additional network interfaces to the system. FP 9300 can handle two single-wide network modules or one double-wide network module.

[2] Also known as the Cisco FXOS chassis.

Figure 1: FP 9300 (first) and FP 4100 (second)

The hardware components in the Firepower appliances (the same FXOS and FTD images run on all the model platforms) have the following distinct characteristics:

Table 5: Firepower 4100 Models

Model	4112	4115	4125	4145
Processor	Intel Xeon Silver 4116 (Skylake)	Intel Xeon Silver 4116 (Skylake)	Intel Xeon Gold 6130 (Skylake)	Intel Xeon Gold 6152 (Skylake)
Crypto Accelerator	CN3550 (NITROX III series die)	CN5560 (NITROX-V GC)	CN5560 (NITROX-V GC)	CN5560 (NITROX-V GC)
Supervisor Blade Processor	Intel i3-3115C (Ivy Bridge)	Intel i3-3115C (Ivy Bridge)	Intel i3-3115C (Ivy Bridge)	Intel i3-3115C (Ivy Bridge)
Storage	400 GB	400 GB	800 GB	800 GB
Integrated network management ports	1 Gigabit Ethernet (Supports 1-G fiber or copper SFPs)			
Serial port	1 x RJ-45 console			

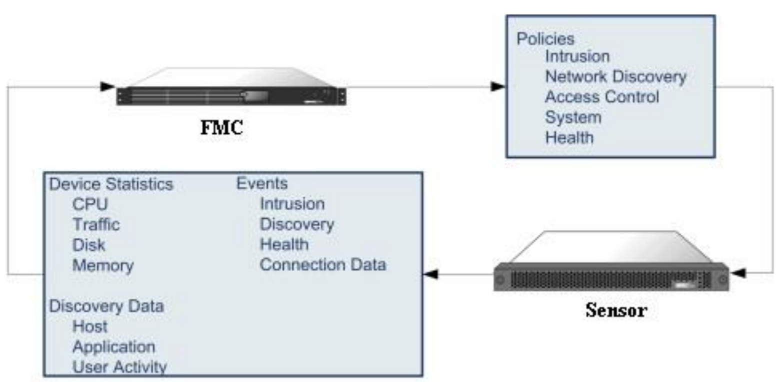
The Firepower eXtensible Operating System (FXOS) is used to manage the FTD. All the platforms run an instance of FXOS that provides management of the hardware and loads FTD. The 4100/9300 chassis runs on its supervisor engine a fully featured build of FXOS referred to as the Management Input Output (MIO) build of FXOS. A separate, more limited build of FXOS runs on any Security Module (SM) installed within the chassis (the Firepower 4100 models contain one fixed Security Module, while the Firepower 9300 chassis supports up to three removable Security Modules). The SM hardware is a form of Cisco UCS server (based on a UCS B-series blade server), and as such it includes a Cisco Integrated Management Controller (CIMC), which is firmware running on a CIMC daughterboard on the server blade. The FTD software runs on FXOS on the SM. The FXOS software running on the chassis supervisor maintains a list of administrative accounts that are able to log in to the supervisor via CLI or WebUI/GUI, called Firepower Chassis Manager (FCM). All administrative accounts can be managed via both CLI and GUI, and the same authentication mechanisms can be used at the CLI or GUI.

The FMC is a fault-tolerant, purpose-built network appliance that provides a centralized management console and database repository for the Sensors (i.e., FTD). The FMC is a key component in the Cisco NGIPS system. Administrators can use the FMC to manage the full range of Sensors that comprise the Cisco NGIPS system, and to aggregate, analyze, and respond to the threats they detect on their network. By using the FMC to manage Sensors, administrators can:

- Configure policies for all Sensors from a single location, making it easier to change configurations.
- Install various types of software updates on Sensors.
- Push policies to managed Sensors and monitor their health status from the FMC.

The FMC aggregates and correlates intrusion events, anomaly, network discovery information, and Sensor performance data, allowing administrators to monitor the information the Sensors are reporting in relation to one another, and to assess the overall activity occurring on their network. The following illustration lists what is transmitted between a FMC and its managed Sensors.

Figure 2: FMC and Managed Sensor



The FMC hardware components in the TOE have the following distinct characteristics:

Table 6: FMC Models

Model	FMC1600	FMC1700	FMC2600	FMC2700	FMC4600	FMC4700
Processor	Intel Xeon Silver 4215 (Cascade Lake)	AMD EPYC 7232P (Zen 2)	Intel Xeon Silver 4215 (Cascade Lake)	AMD EPYC 7282 (Zen 2)	Intel Xeon Silver 4214 (Cascade Lake)	AMD EPYC 7352 (Zen 2)
Memory	32 GB	32 GB	64 GB	64 GB	128 GB	128 GB
Maximum Number of FTD devices Managed	50	50	300	300	750	1000
Event Storage	900 GB	2.4 TB	1.8 TB	2.4 TB	3.2 TB	12 TB
Maximum Flow Rate	5,000 fps	5,000 fps	12,000 fps	12,000 fps	20,000 fps	30,000 fps
Maximum Network Map (hosts/users)	50,000/50,000	50,000/50,000	150,000/150,000	150,000/150,000	600,000/600,000	600,000/600,000
Network Interfaces	Intel X710-DA2 dual-port 10G SFP+ NIC Cisco FMC X710-DA2 dual-port 10G SFP+ NIC	Intel X710-DA2 dual-port 10G SFP+ NIC Intel X710T2LOCPV3G1L 2x10GbE RJ45 OCP3.0 NIC	Intel X710-DA2 dual-port 10G SFP+ NIC Cisco FMC X710-DA2 dual-port 10G SFP+ NIC	Intel X710-DA2 dual-port 10G SFP+ NIC Intel X710T2LOCPV3G1L 2x10GbE RJ45 OCP3.0 NIC	Intel X710-DA2 dual-port 10G SFP+ NIC Cisco FMC X710-DA2 dual-port 10G SFP+ NIC	Intel E810XXVDA2 2x25/10 GbE SFP PCIe NIC Intel X710T2LOCPV3G1L 2x10GbE RJ45 OCP3.0 NIC

The UCS hardware components, which provide the platform for the FMCv, in the TOE have common hardware characteristics. These differing characteristics affect only non-TSF relevant functionality (such as throughput, processing speed, number and type of network connections supported, number of concurrent connections supported, and amount of storage) and therefore support security equivalency of the FMCv in terms of hardware.

Figure 3: UCS Hardware

The UCS hardware components, which provide the platform for the FMCv, in the TOE have the following distinct characteristics:

Table 7: UCS Hardware

Model	C220 M5	C240 M5	C480-M5
Number of Processors	2	2	2-4
Processor	Intel® Xeon® Bronze 3104 (Skylake), Intel® Xeon® Silver 4110 (Skylake) Intel® Xeon® Gold 6128 (Skylake) Intel® Xeon® Platinum 8153 (Skylake)	Intel® Xeon® Bronze 3104 (Skylake), Intel® Xeon® Silver 4110 (Skylake) Intel® Xeon® Gold 6128 (Skylake) Intel® Xeon® Platinum 8153 (Skylake)	Intel® Xeon® Bronze 3104 (Skylake), Intel® Xeon® Silver 4110 (Skylake) Intel® Xeon® Gold 6128 (Skylake) Intel® Xeon® Platinum 8153 (Skylake)
Form factor	1RU rack server	2 RU	4 RU
Maximum Memory	3 TB, 24 x DDR4 DIMMs	3 TB, 24 x DDR4 DIMMs	12 TB, 256 x DDR4 DIMMs
Embedded Network Interface Cards (NICs)	Dual 10GBASE-T Intel x550 Ethernet ports	Dual 10GBASE-T Intel x550 Ethernet ports	Dual 10GBASE-T Intel x550 Ethernet ports

Table 8: UCS M6 Rack Servers

Model	C220 M6	C225 M6	C240 M6
Number of Processors	1 or 2	1 or 2	1 or 2
Processor	Intel Xeon Silver 4310 (Ice Lake) Intel Xeon Silver 4314 (Ice Lake) Intel Xeon Silver 4316 (Ice Lake) Intel Xeon Gold 5315Y (Ice Lake) Intel Xeon Gold 5318N (Ice Lake) Intel Xeon Gold 6312U (Ice Lake) Intel Xeon Gold 6342 (Ice Lake)	AMD EPYC 7232P (Zen 2) AMD EPYC 7252 (Zen 2) AMD EPYC 7262 (Zen 2) AMD EPYC 7272 (Zen 2) AMD EPYC 7282 (Zen 2) AMD EPYC 72F3 (Zen 3) AMD EPYC 7302 (Zen 2)	Intel Xeon Silver 4310 (Ice Lake) Intel Xeon Silver 4314 (Ice Lake) Intel Xeon Silver 4316 (Ice Lake) Intel Xeon Gold 5315Y (Ice Lake) Intel Xeon Gold 5318N (Ice Lake) Intel Xeon Gold 6312U (Ice Lake) Intel Xeon Gold 6342 (Ice Lake)

	Intel Xeon Platinum 8351N (Ice Lake)	AMD EPYC 7313 (Zen 3) AMD EPYC 7343 (Zen 3) AMD EPYC 7352 (Zen 2) AMD EPYC 7373X (Zen 3) AMD EPYC 73F3 (Zen 3) AMD EPYC 7402 (Zen 2) AMD EPYC 74F3 (Zen 3)	Intel Xeon Platinum 8351N (Ice Lake)
Form factor	1RU rack server	1RU rack server	2 RU rack server
Memory	32 DDR4 DIMM slots: 16, 32, 64 and 128 GB up to 3200 MHz	32 DDR4 DIMM slots: 16, 32, 64, 128 and 256 GB up to 3200 MHz	32 DDR4 DIMM slots: 16, 32, 64, 128 and 256 GB up to 3200 MHz
Embedded Network Interface Cards (NICs)	Dual 10GBASE-T Intel x550 Ethernet ports 1 x Modular LOM (mLOM) / OCP Dedicated OOB management port	1 x Modular LOM (mLOM) / OCP Dedicated OOB management port	Dual 10GBASE-T Intel x550 Ethernet ports 1 x Modular LOM (mLOM) / OCP Dedicated OOB management port

Table 9: UCS M7 Rack Servers

Model	C220 M7	C240 M7
Number of Processors	1 or 2	1 or 2
Processor	Intel Xeon Bronze 3408U (Sapphire Rapids) Intel Xeon Silver 4410T (Sapphire Rapids) Intel Xeon Gold 5411N (Sapphire Rapids) Intel Xeon Gold 6414U (Sapphire Rapids)	Intel Xeon Bronze 3408U (Sapphire Rapids) Intel Xeon Silver 4410T (Sapphire Rapids) Intel Xeon Gold 5411N (Sapphire Rapids) Intel Xeon Gold 6414U (Sapphire Rapids)

	Intel Xeon Platinum 8444H (Sapphire Rapids)	Intel Xeon Platinum 8444H (Sapphire Rapids)
Form factor	1RU rack server	2 RU rack server
Memory	32 DDR5 DIMM slots: 16, 32, 64 and 128 GB up to 4800 MT/s	32 DDR5 DIMM slots: 16, 32, 64, 128, and 256 GB up to 4800 MT/s
Embedded Network Interface Cards (NICs)	Intel X710 OCP Dual 10GBase-T via mLOM interposer	Intel X710 OCP Dual 10GBase-T via mLOM interposer

Table 10: UCS E-Series Servers

Model	**E1100D-M6**
Number of Processors	1
Processor	Intel Xeon D-1746TER (Ice lake) Intel Xeon D-2796TE (Ice Lake)
Form factor	2 RU rack server
Memory	32 to 128 GB; 4 DIMM slots, each with 16 GB or 32 GB VLP DDR4 RAM
Disk Space	4 SFF slots; up to 4 TB each (SATA SSD or NVME SSD)
Embedded Network Interface Cards (NICs)	2 internal Gigabit Ethernet ports (Broadcom 5719) 2 external 10 Gigabit Ethernet ports (1000/10000) (integrated within Intel CPU) 1 dedicated management Ethernet port (10/100/1000) for Cisco IMC

1.4 TOE Evaluated Configuration

The TOE consists of at least one Firepower device (Firepower 4100/9300 series) running the FXOS and FTD software and one physical FMC device running the FMC software or virtual devices running FMCv software specified in section 1.5 below. The TOE includes the Cisco FTD, FMC, and FXOS software. Each instantiation of the TOE has two or more network interfaces and is able to filter IP traffic to and through those interfaces.

The TOE can optionally connect to an NTP server via an IPsec tunnel for clock updates. If the TOE is to be remotely administered, the management station must connect using SSHv2. When web UI is used, a remote workstation with a TLS-enabled browser must be available. A syslog server can also be used to store audit records, and the syslog server must support syslog over TLS or IPsec.

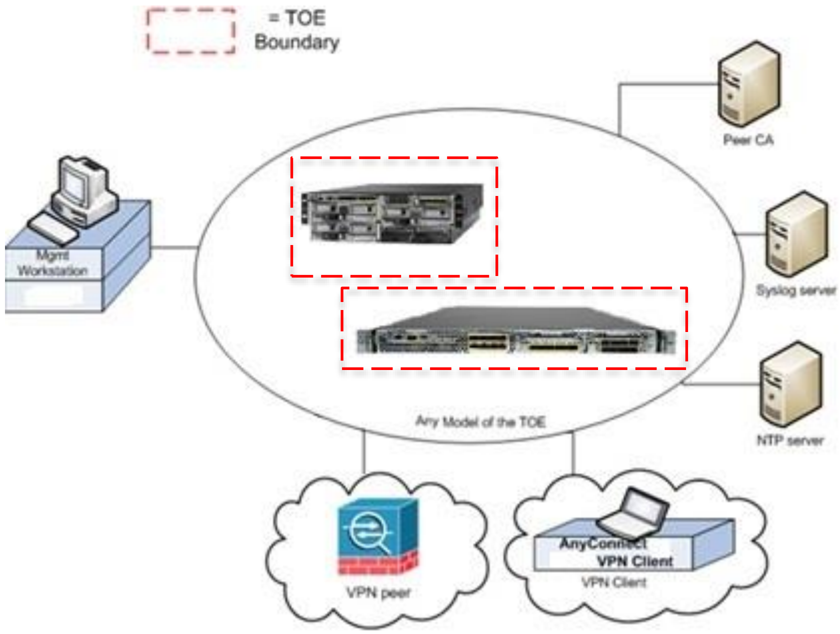
FTD supports two different TLS clients that send syslog messages to the external syslog server- FTD TLS client and FTD OS TLS Client. The FTD TLS Client is configured by the FMC and is the main audit system for audits generated by FTD. It sends audit events such as IPsec and login messages to the external syslog server. Mutual authentication is not supported. The FTD OS TLS client implementation is configured through the FTD's command line and sends audit events such as SSH login, console login, etc. to an external syslog server. Mutual authentication is not supported.

The TOE can filter connections to/from these external entities using its IP traffic filtering, and can encrypt traffic where necessary using TLS, SSH, and/or IPsec. The TOE uses X.509v3 certificates to support authentication for both IPsec and TLS, and the CA server in the Operational environment can be used to obtain digital certificates.

The communication between the FMC software and FTD in Firepower appliance is protected by TLSv1.2. Digital certificates from a CA server are obtained when certificates are used as the authentication method for VPN connection. The TOE protects peer-to-peer VPN connections between itself and VPN peers (connections can be initiated by the TOE or by the peer) using IPsec.

The following figure provides a visual depiction of an example TOE deployment. The TOE boundary is surrounded with a hashed red line.

Figure 4: Example TOE Deployment



The previous figure includes the following:

TOE components (at least one Firepower 4100/9300 appliance (with FTD and FXOS) and FMC)
VPN Peer (Operational Environment) or another instance of the TOE
VPN Client (Operational Environment) (Cisco AnyConnect VPN Client)
Management Workstation (Operational Environment)
NTP Server (Operational Environment)
CA Server (Operational Environment)
Syslog server (Operational Environment)

1.5 Physical Scope of the TOE

The TOE is a hardware and software solution comprised of the components described in Table 11:

Table 11: Hardware Models and Specifications

TOE Configuration	Hardware Configurations	Software Version
FP 4112 **FP 4115** **FP 4125** **FP 4145**	The Firepower 4100 chassis contains the following components: • Network module 1 with eight fixed SFP+ ports (1G and 10G connectivity), the management port, RJ-45 console port, Type A USB port, PID and S/N card, locator indicator, and power switch	FXOS release 2.14 and FTD release 7.4

	• Two network modules slots (network module 2 and network module 3) • Two (1+1) redundant power supply module slots • Six fan module slots • Two SSD bays	
FP 9300	The Firepower 9300 chassis contains the following components: • Firepower 9300 Supervisor—Chassis supervisor module ◦ Management port ◦ RJ-45 console port ◦ Type A USB port ◦ Eight ports for 1 or 10 Gigabit Ethernet SFPs (fiber and copper) • Firepower 9300 Security Module—Up to three security modules ◦ 800 GB of solid state storage per security blade (2 x 800 GB solid state drives running RAID1) • Firepower Network Module—Two single-wide network modules or one double-wide network module • Two power supply modules (AC or DC) • Four fan modules	FXOS release 2.14 and FTD release 7.4
FMC1600 **FMC2600** **FMC4600** **FMC1700** **FMC2700** **FMC4700**	See Table 6	FMC release 7.4
FMCv	FMCv running on ESXi 7.0 on the Unified Computing System (UCS) UCSC-C220-M5, UCSC-C240-M5, UCSC-C480-M5, UCSC-C220-M6, UCSC-C225-M6, UCSC-C240-M6, UCSC-C220-M7, UCSC-C240-M7and UCS-E1100D-M6	FMCv release 7.4

1.6 Logical Scope of the TOE

The TOE is comprised of several security features including stateful traffic firewall, VPN gateway and IPS capabilities. Each of the security features identified above consists of several security functionalities, as identified below.

1. Security Audit
2. Communication
3. Cryptographic Support
4. User Data Protection
5. Identification and Authentication
6. Security Management
7. Protection of the TSF
8. TOE Access
9. Trusted Path/Channels
10. Filtering
11. Intrusion Prevention System

These features are described in more detail in the subsections below -

1.6.1 Security Audit

The TOE provides extensive auditing capabilities. The TOE can audit events related to cryptographic functionality, identification and authentication, and administrative actions. The TOE generates an audit record for each auditable event. The administrator configures auditable events, performs back-up operations, and manages audit data storage. The TOE provides the administrator with a circular audit trail where the TOE overwrites the oldest audit record with the newest audit record when space is full. Audit logs are backed up over an encrypted channel to an external audit server.

1.6.2 Communication

The TOE allows authorized administrators to control which FTD device is managed by the FMC. This is performed through a registration process over TLS. The administrator can also de-register a FTD device if he or she wish to no longer manage it through the FMC.

1.6.3 Cryptographic Support

The TOE provides cryptography in support of other TOE security functionality. The TOE provides cryptography in support of secure connections using IPsec and TLS, and remote administrative management via SSHv2, and TLS/HTTPS. The cryptographic random bit generators (RBGs) are seeded by a platform-based entropy noise source.

1.6.4 User Data Protection

The TOE ensures that all information flows from the TOE do not contain residual information from previous traffic. Packets are padded with zeros. Residual data is never transmitted from the TOE.

1.6.5 Identification and authentication

The TOE performs two types of authentication: device-level authentication of the remote device (VPN peers) and user authentication for the authorized administrator of the TOE or for IPsec VPN clients. Device-level authentication allows the TOE to establish a secure channel with a trusted peer. The secure channel is established only after each device authenticates the other. Device-level authentication is performed via IKE/IPsec X509v3 certificate based authentication while user-level authentication from IPsec VPN clients uses certificate-based authentication (all IPsec VPN sessions are terminated at the FTD, not the FMC/FMCv).

The TOE provides authentication services for administrative users wishing to connect to the TOEs secure CLI and GUI administrator interfaces. The TOE requires authorized administrators to authenticate prior to being granted access to any of the management functionality. The TOE can be configured to require a minimum password length between 1 and 127 characters for FTD, 8 and 127 characters for FMC and FXOS as well as mandatory password complexity rules. The TOE also implements a lockout mechanism when the number of unsuccessful authentication attempts exceeds the configured threshold.

The TOE provides administrator authentication against a local user database. Password-based authentication can be performed on the serial console or SSH and HTTPS interfaces. The SSHv2 interface also supports authentication using SSH keys.

1.6.6 Security Management

The TOE provides secure administrative services for management of general TOE configuration and the security functionality provided by the TOE. All TOE administration occurs either through a secure SSHv2 or TLS/HTTPS session, or via a local console connection. Optionally, the FXOS and FTD support tunneling the SSH and HTTPS connections in IPsec VPN tunnels (remote VPN client). Management of all security functions can be performed via the FMC/FMCv component of the TOE, while a subset of management functions can be performed on the FTD and FXOS. The TOE provides the ability to securely manage all TOE administrative users; all identification and authentication; all audit functionality of the TOE; all TOE cryptographic functionality; the timestamps maintained by the TOE; and the information flow control policies enforced by the TOE including encryption/decryption of information flows for VPNs. The TOE supports an “authorized administrator” role, which equates to any account authenticated to an administrative interface (CLI or GUI, but not VPN), and possessing sufficient privileges to perform security-relevant administrative actions.

When an administrative session is initially established, the TOE displays an administrator- configurable warning banner. This is used to provide any information deemed necessary by the administrator. After a configurable period of inactivity, administrative sessions will be terminated, requiring administrators to re-authenticate.

1.6.7 Protection of the TSF

The TOE protects against interference and tampering by untrusted subjects by implementing identification, authentication, and administrator roles to limit configuration to authorized administrators. The TOE prevents reading of cryptographic keys and passwords.

Additionally, the TOE is not a general-purpose operating system and access to the TOE memory space is restricted to only TOE functions.

The TOE internally maintains the date and time. This date and time are used as the timestamp that is applied to audit records generated by the TOE. Administrators can update the TOE's clock manually via FMC or FXOS or can configure the TOE (FXOS) to use NTP via an IPsec tunnel to synchronize the TOE's clock with an external time source. Additionally, the TOE performs testing to verify correct operation of the appliance itself and that of the cryptographic module. Whenever any system failures occur within the TOE the TOE will cease operation.

The TOE provides the ability to manually upgrade firmware/software for security administrators. Administrators can query the current executing version of the TOE's firmware/software and the most recently installed version via the FMC Web UI.

1.6.8 TOE Access

When an administrative session is initially established, the TOE displays an administrator- configurable warning banner. This is used to provide any information deemed necessary by the administrator. After a configurable period of inactivity, administrator and VPN client sessions will be terminated, requiring re-authentication. The TOE also supports direct connections from VPN clients and protects against threats related to those client connections. The TOE disconnects sessions that have been idle too long and can be configured to deny sessions based on IP, time, and day, and to NAT external IPs of connecting VPN clients to internal network addresses.

1.6.9 Trusted path/Channels

The TOE supports establishing trusted paths between itself and remote administrators using SSHv2 for CLI access (FMC, FTD, FXOS), and TLS/HTTPS for GUI access (FMC, FXOS). The TOE supports use of TLS and/or IPsec for connections with remote syslog servers and use of IPsec for connections with NTP servers. The TOE can establish trusted paths of peer-to-peer VPN tunnels using IPsec, and VPN client tunnels using IPsec or TLS. Note that the VPN client is in the operational environment.

1.6.10 Filtering

The TOE provides stateful traffic firewall functionality including IP address-based filtering (for IPv4 and IPv6) 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. Address filtering can be configured to restrict the flow of network traffic between protected networks and other attached networks based on source and/or destination IP addresses. Port filtering can be configured to restrict the flow of network traffic between protected networks and other attached networks based on the originating (source) and/or receiving (destination) port (service). Stateful packet inspection is used to aid in the performance of packet flow through the TOE and to ensure that only packets are only forwarded when they're part of a properly established session. The TOE supports protocols that can spawn additional sessions in accordance with the protocol RFCs where a new connection will be implicitly permitted when properly initiated by an explicitly permitted session. The File Transfer Protocol is an example of such a protocol, where a data connection is created as needed in response to an explicitly allowed command connection. System monitoring functionality includes the ability to generate audit messages for any explicitly defined (permitted or denied) traffic flow. TOE administrators have the ability to configure permitted and denied traffic flows, including adjusting the sequence in which flow control rules will be applied, and to apply rules to any network interface of the TOE.

The TOE also provides packet filtering and secure IPsec tunneling. The tunnels can be established between two trusted VPN peers as well as between remote VPN clients and the TOE. More accurately, these tunnels are sets of security associations (SAs). The SAs define the protocols and algorithms to be applied to sensitive packets and specify the keying material to be used. SAs are unidirectional and are established per the ESP security protocol. An authorized administrator can define the traffic that needs to be protected via IPsec by configuring access lists (permit, deny, log) and applying these access lists to interfaces using VPN policies.

1.6.11 Intrusion Prevention System

The TOE provides intrusion policies consisting of rules and configurations invoked by the access control policy. The intrusion policies are the last line of defense before the traffic is allowed to its destination. All traffic permitted by the access control policy is then inspected by the designated intrusion policy. Using intrusion rules and other preprocessor settings, these policies inspect traffic for security violations and, in inline deployments, can block or alter malicious traffic.

If the vendor-provided intrusion policies do not fully address the security needs of the organization, custom policies can improve the performance of the system in the environment and can provide a focused view of the malicious traffic and policy violations occurring on the network. By creating and tuning custom policies, the administrators can configure, at a very granular level, how the system processes and inspects the traffic on the network for intrusions.

Using Security Intelligence, the administrators can blacklist—deny traffic to and from—specific IP addresses, URLs, and DNS domain names, before the traffic is subjected to analysis by the access control rules. Optionally, the administrators can use a “monitor-only” setting for Security Intelligence filtering.

1.7 Excluded Functionality

The following functionality is excluded from the evaluation.

Table 12: Excluded Functionality

Excluded Functionality	Exclusion Rationale
Telnet for management purposes	Telnet passes authentication credentials in clear text and is disabled by default.
Firepower Device Manager (FDM)	Firepower Device Manager is a web-based local manager. Use of FDM is beyond the scope of this Common Criteria evaluation.
Filtering of non-IP traffic provided by the EtherType option when configuring information flow policies is excluded from the evaluated configuration	Use of non-IP traffic filtering is beyond the scope of this Common Criteria evaluation.
Smart Call Home. The Smart Call Home feature provides personalized, e-mail-based and web-based notification to customers about critical events involving their individual systems.	Use of Smart Call Home is beyond the scope of this Common Criteria evaluation.

Root Shell Access	The root shell access is only allowed for pre-operational installation, configuration, and post-operational maintenance and troubling shooting.
Timeout Exemption Option	The use of the “Exempt from Browser Session Timeout” setting is not permitted. This allows a user to be exempted from the inactivity timeout feature.
FXOS REST API	Allows users to programmatically configure and manage their chassis. Use of REST API is beyond the scope of this Common Criteria evaluation.
Clustering	This feature is not tested and is out of scope.

The services in the table above are disabled in the evaluated configuration. Any functionality of the TOE that has not been discussed in Section 6 of this document is not included in the evaluation. The exclusion of this functionality does not affect compliance to collaborative Protection Profile for Network Devices (cpp_nd_v2.2e), PP-Module for Stateful Traffic Filter Firewalls (mod_cpp_fw_v1.4e), PP-Module for Virtual Private Network (VPN) Gateways (mod_vpngw_v1.3) and PP-Module for Intrusion Protection Systems (IPS), Version 1.0 (MOD_IPS_V1.0)

2 CONFORMANCE CLAIMS

2.1 Common Criteria Conformance Claim

The TOE and ST are compliant with the Common Criteria (CC) Version 3.1, Revision 5, dated: April 2017. For a listing of Assurance Requirements claimed see section 5.8.

The TOE and ST are CC Part 2 extended and CC Part 3 conformant.

2.2 Protection Profile Conformance

The TOE and ST are conformant with the Protection Profiles as listed in Table 13 below:

Table 13: Protection Profiles

Protection Profile	Version	Date
PP-Configuration for Network Device, Intrusion Prevention Systems (IPS), Stateful Traffic Filter Firewalls, and Virtual Private Network (VPN) Gateways (CFG_NDcPP-IPS-FW-VPNGW_V1.2)	1.2	18 August 2023
The PP-Configuration includes the following components:		
• Base-PP: Collaborative Protection Profile for Network Devices, (CPP_ND_V2.2E)	2.2e	23 March 2020
• PP-Module: PP-Module for Intrusion Prevention Systems (IPS), (MOD_IPS_V1.0)	1.0	11 May 2021
• PP-Module for Stateful Traffic Filter Firewalls, (MOD_CPP_FW_1.4E)	1.4 + Errata 20200625	25 June 2020
• PP-Module for Virtual Private Network (VPN) Gateways, (MOD_VPNGW_V1.3)	1.3	16 August 2023

The TOE and ST are conformant with the Protection Profiles as listed in Table above. The following NIAP Technical Decisions (TD) have also been applied:

Table 14: Technical Decisions

TD #	TD Name	Protection Profiles	Applied to this TOE
TD0878	Updating FIPS 186-4 to 186-5 in MOD_VPNGW_V1.3	MOD_VPNGW_v1.3	FCS_CKM.1/IKE
TD0838	*PPK Configurability in FIA_PSK_EXT.1.1*	*MOD_VPNGW_v1.3*	*Not applied because this ST does not include FIA_PSK_EXT.1*
TD0828	*Aligning MOD_IPS_V1.0 with CPP_ND_V3.0E*	*MOD_IPS_V1.0*	*Not applied because the base PP of this evaluation is NDcPPv2.2e*
TD0827	*Aligning MOD_CPP_FW_v1.4E with CPP_ND_V3.0E*	*MOD_CPP_FW_v1.4e*	*Not applied because the base PP of this evaluation is NDcPPv2.2e*

TD0824	*Aligning MOD_VPNGW 1.3 with NDcPP 3.0E*	*MOD_VPNGW_v1.3*	*Not applied because the base PP of this evaluation is NDcPPv2.2e*
TD0811	*Correction to Referenced SFR in FIA_PSK_EXT.3 Test*	*MOD_VPNGW_v1.3*	*Not applied because this ST does not include FIA_PSK_EXT.3*
TD0800	Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	CPP_ND_V2.2E	FCS_IPSEC_EXT.1.7, FCS_IPSEC_EXT.1.8
TD0792	NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	CPP_ND_V2.2E	FIA_PMG_EXT.1
TD0790	NIT Technical Decision: Clarification Required for testing IPv6	CPP_ND_V2.2E	FCS_TLSC_EXT.1.2
TD0781	*Correction to FIA_PSK_EXT.3 EA for MOD_VPNGW_v1.3*	*MOD_VPNGW_v1.3*	*Not applied because this ST does not include FIA_PSK_EXT.3*
TD0738	NIT Technical Decision for Link to Allowed-With List	CPP_ND_V2.2E	Section 2 of PP
TD0722	IPS_SBD_EXT.1.1 EA Correction	MOD_IPS_V1.0	IPS_SBD_EXT.1.1
TD0670	NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	CPP_ND_V2.2E	FCS_TLSC_EXT.2.1
TD0639	NIT Technical Decision for Clarification for NTP MAC Keys	CPP_ND_V2.2E	FCS_NTP_EXT.1.2, FAU_GEN.1, FCS_CKM.4, FPT_SKP_EXT.1
TD0638	NIT Technical Decision for Key Pair Generation for Authentication	CPP_ND_V2.2E	FCS_CKM.1
TD0636	*NIT Technical Decision for Clarification of Public Key User Authentication for SSH*	*CPP_ND_V2.2E*	*Not applied because this ST does not include FCS_SSHC_EXT.1*
TD0635	NIT Technical Decision for TLS Server and Key Agreement Parameters	CPP_ND_V2.2E	FCS_TLSS_EXT.1.3
TD0632	NIT Technical Decision for Consistency with Time Data for vNDs	CPP_ND_V2.2E	FPT_STM_EXT.1.2
TD0631	NIT Technical Decision for Clarification of public key authentication for SSH Server	CPP_ND_V2.2E	FCS_SSHS_EXT.1, FMT_SMF.1
TD0595	Administrative corrections to IPS PP-Module	MOD_IPS_V1.0	FAU_GEN.1.1/IPS
TD0592	NIT Technical Decision for Local Storage of Audit Records	CPP_ND_V2.2E	FAU_STG
TD0591	NIT Technical Decision for Virtual TOEs and hypervisors	CPP_ND_V2.2E	A.LIMITED_FUNCTIONALITY
TD0581	NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	CPP_ND_V2.2E	FCS_CKM.2
TD0580	NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	CPP_ND_V2.2E	FCS_CKM.1.1, FCS_CKM.2.1
TD0572	NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	CPP_ND_V2.2E	FTP_ITC.1
TD0571	NiT Technical Decision for Guidance on how to handle FIA_AFL.1	CPP_ND_V2.2E	FIA_AFL.1
TD0570	NiT Technical Decision for Clarification about FIA_AFL.1	CPP_ND_V2.2E	FIA_AFL.1
TD0569	NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	CPP_ND_V2.2E	FCS_TLSS_EXT.1

TD0564	NiT Technical Decision for Vulnerability Analysis Search Criteria	CPP_ND_V2.2E	AVA_VAN.1
TD0563	NiT Technical Decision for Clarification of audit date information	CPP_ND_V2.2E	FAU_GEN.1
TD0556	NIT Technical Decision for RFC 5077 question	CPP_ND_V2.2E	FCS_TLSS_EXT.1
TD0555	NIT Technical Decision for RFC Reference incorrect in TLSS Test	CPP_ND_V2.2E	FCS_TLSS_EXT.1
TD0551	NIT Technical Decision for Incomplete Mappings of OEs in FW Module v1.4+Errata	MOD_CPP_FW_v1.4e	Sections 5.3.2 and 5.3.4
TD0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	CPP_ND_V2.2E	AVA_VAN.1
TD0546	NIT Technical Decision for DTLS - clarification of Application Note 63	CPP_ND_V2.2E	Not applied because this ST does not include FCS_DTLSC_EXT.1.1
TD0545	NIT Technical Decision for Conflicting FW rules cannot be configured (extension of RfI#201837)	MOD_CPP_FW_v1.4e	FFW_RUL_EXT.1.8
TD0537	NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	CPP_ND_V2.2E	FCS_TLSC_EXT.2.3
TD0536	NIT Technical Decision for Update Verification Inconsistency	CPP_ND_V2.2E	AGD_OPE.1
TD0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	CPP_ND_V2.2E	FCS_NTP_EXT.1.4
TD0527	Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	CPP_ND_V2.2E	FIA_X509_EXT.1/REV, FIA_X509_EXT.1/ITT

2.3 Protection Profile Conformance Claim Rationale

2.3.1 TOE Appropriateness

The TOE provides all the functionality at a level of security commensurate with that identified in the:

- collaborative Protection Profile for Network Devices (cpp_nd_v2.2e)
- PP-Module: PP-Module for Intrusion Protection Systems (IPS), (mod_ips_v1.0)
- PP-Module for Stateful Traffic Filter Firewalls (mod_cpp_fw_v1.4e); and
- PP-Module for Virtual Private Network (VPN) Gateways (mod_vpngw_v1.3)

2.3.2 TOE Security Problem Definition Consistency

The Assumptions, Threats, and Organization Security Policies included in the Security Target represent the Assumptions, Threats, and Organization Security Policies specified in the NDcPPv2.2e, mod_ips_v1.0, mod_cpp_fw_v1.4e and mod_vpngw_v1.3 for which conformance is claimed verbatim. All concepts covered in the Protection Profile Security Problem Definition are included in the Security Target Statement of Security Objectives Consistency.

The Security Objectives included in the Security Target represent the Security Objectives specified in the U.S. Government Protection Profile for Security Requirements for Network Devices for which conformance is claimed verbatim. All concepts covered in the Protection Profile's Statement of Security Objectives are included in the Security Target.

2.3.3 Statement of Security Requirements Consistency

The Security Functional Requirements included in the Security Target represent the Security Functional Requirements specified in cpp_nd_v2.2e, mod_ips_v1.0, mod_cpp_fw_v1.4e and mod_vpngw_v1.3 for which conformance is claimed verbatim and several additional Security Functional Requirements are included as a result. All concepts covered the Protection Profile's Statement of Security Requirements are included in the Security Target. Additionally, the Security Assurance Requirements included in the Security Target are identical to the Security Assurance Requirements included in section 7 of the NDcPP.

3 SECURITY PROBLEM DEFINITION

This chapter identifies the following:

- Significant assumptions about the TOE's operational environment.
- IT related threats to the organization countered by the TOE.
- Environmental threats requiring controls to provide sufficient protection.
- Organizational security policies for the TOE as appropriate.

This document identifies assumptions as A.assumption with "assumption" specifying a unique name. Threats are identified as T.threat with "threat" specifying a unique name. Organizational Security Policies (OSPs) are identified as P.osp with "osp" specifying a unique name.

3.1 Assumptions

The specific conditions listed in the following subsections are assumed to exist in the TOE's environment. These assumptions include both practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

Table 15: TOE Assumptions

Assumption	**Assumption Definition**
Reproduced from cpp_nd_v2.2e	
A.PHYSICAL_PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP 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 device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.
A.LIMITED_FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality). If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform.
A.NO_THRU_TRAFFIC_PROTECTION	A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the

Assumption	Assumption Definition
	device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the ND cPP. It is assumed that this protection will be covered by cPPs and PP modules for particular types of Network Devices (e.g., firewall).
A.TRUSTED_ADMINSTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes 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 device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
A.REGULAR_UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_ SECURE	The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.
A.COMPONENTS_RUNNING	For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.
A.VS_TRUSTED_ADMINISTRATOR	The Security Administrators for the VS are assumed to be trusted and to act in the best interest of security for the organization. This includes not interfering with the correct operation of the device. The Network Device is not expected to be capable of defending against a malicious VS Administrator that actively works to bypass or compromise the security of the device.

Assumption	Assumption Definition
A.VS_REGULAR_UPDATES	The VS software is assumed to be updated by the VS Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.VS_ISOLATON	For vNDs, it is assumed that the VS provides, and is configured to provide sufficient isolation between software running in VMs on the same physical platform. Furthermore, it is assumed that the VS adequately protects itself from software running inside VMs on the same physical platform.
A.VS_CORRECT_CONFIGURATION	For vNDs, it is assumed that the VS and VMs are correctly configured to support ND functionality implemented in VMs.
Reproduced from mod_ips_v1.0 and mod_vpngw_v1.3	
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.

3.2 Threats

The following table lists the threats addressed by the TOE and the IT Environment. The assumed level of expertise of the attacker for all the threats identified below is Enhanced-Basic.

Table 16: Threats

Threat	Threat Definition
Reproduced from cpp_nd_v2.2e	
T.UNAUTHORIZED_ ADMINISTRATOR_ACCESS	Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.

Threat	Threat Definition
T.UNTRUSTED_COMMUNICATIONS _CHANNELS	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.
T.WEAK_AUTHENTICATION_ ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints, e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised.
T.UPDATE_COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_FUNCTIONALITY_ COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic and may allow them to take advantage of any trust relationships with other Network Devices.

Threat	Threat Definition
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.
Reproduced from mod_ips_v1.0	
T.NETWORK_DISCLOSURE[IPS]	Sensitive information on a protected network might be disclosed resulting from ingress-or egress-based actions.
T.NETWORK_ACCESS[IPS]	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[IPS]	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[IPS]	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.
Reproduced from the mod_cpp_fw_v1.4e	
T.NETWORK_DISCLOSURE[FW]	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[FW]	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[FW]	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.

Threat	Threat Definition
T.MALICIOUS_TRAFFIC[FW]	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.
Reproduced from the mod_vpngw_v1.3	
T.DATA_INTEGRITY[VPN]	Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to modify the data without authorization. If known malicious external devices are able to communicate with devices on the protected network or if devices on the protected network can establish communications with those external devices then the data contained within the communications may be susceptible to a loss of integrity.
T. NETWORK_ACCESS[VPN]	Devices located outside the protected network may seek to exercise services located on the protected network that are intended to only be accessed from inside the protected network or only accessed by entities using an authenticated path into the protected network. Devices located outside the protected network may, likewise, offer services that are inappropriate for access from within the protected network. From an ingress perspective, VPN gateways can be configured so that only those network servers intended for external consumption by entities operating on a trusted network (e.g., machines operating on a network where the peer VPN gateways are supporting the connection) are accessible and only via the intended ports. This serves to mitigate the potential for network entities outside a protected network to access network servers or services intended only for consumption or access inside a protected network. From an egress perspective, VPN gateways can be configured so that only specific external services (e.g., based on destination port) can be accessed from within a protected network, or moreover are accessed via an encrypted channel. For example, access to external mail services can be blocked to enforce corporate policies against accessing uncontrolled e-mail servers, or, that access to the mail server must be done over an encrypted link.

Threat	Threat Definition
T. NETWORK_DISCLOSURE[VPN]	Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to conduct unauthorized activities. If known malicious external devices are able to communicate with devices on the protected network, or if devices on the protected network can establish communications with those external devices (e.g., as a result of a phishing episode or by inadvertent responses to email messages), then those internal devices may be susceptible to the unauthorized disclosure of information. From an infiltration perspective, VPN gateways serve not only to limit access to only specific destination network addresses and ports within a protected network, but whether network traffic will be encrypted or transmitted in plaintext. With these limits, general network port scanning can be prevented from reaching protected networks or machines, and access to information on a protected network can be limited to that obtainable from specifically configured ports on identified network nodes (e.g., web pages from a designated corporate web server). Additionally, access can be limited to only specific source addresses and ports so that specific networks or network nodes can be blocked from accessing a protected network thereby further limiting the potential disclosure of information. From an exfiltration perspective, VPN gateways serve to limit how network nodes operating on a protected network can connect to and communicate with other networks limiting how and where they can disseminate information. Specific external networks can be blocked altogether or egress could be limited to specific addresses and/or ports. Alternately, egress options available to network nodes on a protected network can be carefully managed in order to, for example, ensure that outgoing connections are encrypted to further mitigate inappropriate disclosure of data through packet sniffing.

Threat	Threat Definition
T.NETWORK_MISUSE[VPN]	Devices located outside the protected network, while permitted to access particular *public* services offered inside the protected network, may attempt to conduct inappropriate activities while communicating with those allowed public services. Certain services offered from within a protected network may also represent a risk when accessed from outside the protected network. From an ingress perspective, it is generally assumed that entities operating on external networks are not bound by the use policies for a given protected network. Nonetheless, VPN gateways can log policy violations that might indicate violation of publicized usage statements for publicly available services. From an egress perspective, VPN gateways can be configured to help enforce and monitor protected network use policies. As explained in the other threats, a VPN gateway can serve to limit dissemination of data, access to external servers, and even disruption of services – all of these could be related to the use policies of a protected network and as such are subject in some regards to enforcement. Additionally, VPN gateways can be configured to log network usages that cross between protected and external networks and as a result can serve to identify potential usage policy violations.
T.REPLAY_ATTACK[VPN]	If an unauthorized individual successfully gains access to the system, the adversary may have the opportunity to conduct a “replay” attack. This method of attack allows the individual to capture packets traversing throughout the network and send the packets at a later time, possibly unknown by the intended receiver. Traffic is subject to replay if it meets the following conditions: • Cleartext: an attacker with the ability to view unencrypted traffic can identify an appropriate segment of the communications to replay as well in order to cause the desired outcome. • No integrity: alongside cleartext traffic, an attacker can make arbitrary modifications to captured traffic and replay it to cause the desired outcome if the recipient has no means to detect these modifications.

3.3 Organizational Security Policies

The following table lists the Organizational Security Policies imposed by an organization to address its security needs.

Table 17: Organizational Security Policies

Policy Name	Policy Definition
Reproduced from cpp_nd_v2.2e	
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.
Reproduced from mod_ips_v1.0	
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

This section identifies the security objectives of the TOE and the IT Environment. The security objectives identify the responsibilities of the TOE and the TOE's IT environment in meeting the security needs.

- This document identifies objectives of the TOE as O.objective with objective specifying a unique name. Objectives that apply to the IT environment are designated as OE.objective with objective specifying a unique name.

4.1 Security Objectives for the TOE

The following table, Security Objectives for the TOE, identifies the security objectives of the TOE. These security objectives reflect the stated intent to counter identified threats and/or comply with any security policies identified. An explanation of the relationship between the objectives and the threats/policies is provided in the rationale section of this document.

Table 18: Security Objectives for the TOE

TOE Objective	TOE Security Objective Definition
Reproduced from mod_ips_v1.0	
O.IPS_ANALYZE	Entities that reside on or communicate across monitored networks must have network activity effectively analyzed for potential violations of approved network usage. The TOE must be able to effectively analyze data collected from monitored networks to reduce the risk of unauthorized disclosure of information, inappropriate access to services, and misuse of network resources.
O.IPS_REACT	The TOE must be able to react in real-time as configured by the Security Administrator to terminate and block traffic flows that have been determined to violate administrator-defined IPS policies.
O.SYSTEM_MONITORING	To be able to analyze and react to potential network policy violations, the IPS must be able to collect and store essential data elements of network traffic on monitored networks.
O.TOE_ADMINISTRATION	To address the threat of unauthorized administrator access that is defined in the Base-PP, conformant TOEs will provide the functions necessary for an administrator to configure the IPS capabilities of the TOE.
Reproduced from mod_cpp_fw_v1.4e	
O.RESIDUAL_INFORMATION	The TOE shall implement measures to ensure that any previous information content of network packets sent

TOE Objective	TOE Security Objective Definition
	through the TOE is made unavailable either upon deallocation of the memory area containing the network packet or upon allocation of a memory area for a newly arriving network packet or both.
O.STATEFUL_TRAFFIC_FILTERING	The TOE shall perform stateful traffic filtering on network packets that it processes. For this the TOE shall support the definition of stateful traffic filtering rules that allow to permit or drop network packets. The TOE shall support assignment of the stateful traffic filtering rules to each distinct network interface. The TOE shall support the processing of the applicable stateful traffic filtering rules in an administratively defined order. The TOE shall deny the flow of network packets if no matching stateful traffic filtering rule is identified.
Reproduced from mod_vpngw_v1.3	
O.ADDRESS_FILTERING	To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption or denial of services, and network-based reconnaissance, compliant TOE's will implement Packet Filtering capability. That capability will restrict the flow of network traffic between protected networks and other attached networks based on network addresses of the network nodes originating (source) and/or receiving (destination) applicable network traffic as well as on established connection information.
O.AUTHENTICATION	To further address the issues associated with unauthorized disclosure of information, a compliant TOE's authentication ability (IPSec) will allow a VPN peer to establish VPN connectivity with another VPN peer. VPN endpoints authenticate each other to ensure they are communicating with an authorized external IT entity.
O.CRYPTOGRAPHIC_FUNCTIONS	To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption of services, and network-based reconnaissance, compliant TOE's will implement a cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.
O.FAIL_SECURE	There may be instances where the TOE's hardware malfunctions or the integrity of the TOE's software is compromised, the latter being due to malicious or non-malicious intent. To address the concern of the TOE

TOE Objective	TOE Security Objective Definition
	operating outside of its hardware or software specification, the TOE will shut down upon discovery of a problem reported via the self-test mechanism and provide signature-based validation of updates to the TSF.
O.PORT_FILTERING	To further address the issues associated with unauthorized disclosure of information, etc., a compliant TOE's port filtering capability will restrict the flow of network traffic between protected networks and other attached networks based on the originating (source) and/or receiving (destination) port (or service) identified in the network traffic as well as on established connection information.
O.SYSTEM_MONITORING	To address the issues of administrators being able to monitor the operations of the VPN gateway, it is necessary to provide a capability to monitor system activity. Compliant TOEs will implement the ability to log the flow of network traffic. Specifically, the TOE will provide the means for administrators to configure packet filtering rules to 'log' when network traffic is found to match the configured rule. As a result, matching a rule configured to 'log' will result in informative event logs whenever a match occurs. In addition, the establishment of security associations (SAs) is auditable, not only between peer VPN gateways, but also with certification authorities (CAs).
O.TOE_ADMINISTRATION	TOEs will provide the functions necessary for an administrator to configure the packet filtering rules, as well as the cryptographic aspects of the IPsec protocol that are enforced by the TOE.

4.2 Security Objectives for the Environment

All of the assumptions stated in section 3.1 are considered to be security objectives for the environment. The following are the Protection Profile non-IT security objectives, which, in addition to those assumptions, are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software. Thus, they will be satisfied largely through application of procedural or administrative measures.

Table 19: Security Objectives for the Environment

Environment Security Objective	IT Environment Security Objective Definition
Reproduced from cpp_nd_v2.2e	
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE’s trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_ SECURE	The administrator’s credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.COMPONENTS_RUNNING	For distributed TOEs the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Security Administrator also ensures

Environment Security Objective	IT Environment Security Objective Definition
	that it is checked as appropriate for every TOE component that the audit functionality is running properly.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.
OE.VM_CONFIGURATION	For vNDs, the Security Administrator ensures that the VS and VMs are configured to • reduce the attack surface of VMs as much as possible while supporting ND functionality (e.g., remove unnecessary virtual hardware, turn off unused inter-VM communications mechanisms), and • correctly implement ND functionality (e.g., ensure virtual networking is properly configured to support network traffic, management channels, and audit reporting). The VS should be operated in a manner that reduces the likelihood that vND operations are adversely affected by virtualization features such as cloning, save/restore, suspend/resume, and live migration. If possible, the VS should be configured to make use of features that leverage the VS's privileged position to provide additional security functionality. Such features could include malware detection through VM introspection, measured VM boot, or VM snapshot for forensic analysis.
Reproduced from mod_ips_v1.0	
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.
Reproduced from mod_vpngw_v1.3	
OE.CONNECTIONS	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

This section identifies the Security Functional Requirements for the TOE. The Security Functional Requirements included in this section are derived from Part 2 of the *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, dated: April 2017* and all international interpretations.

5.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with *italicized* text;
- Refinement made by PP author: Indicated with **bold** text;
- Selection: Indicated with underlined text;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).
- Where operations were completed in the cpp_nd_v2.2e, mod_ips_v1.0, mod_cpp_fw_v1.4e and mod_vpngw_v1.3 itself, the formatting used there has been retained.

Extended SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs. Formatting conventions outside of operations and iterations matches the formatting specified within the PP and the PP modules themselves. In addition, SFRs copied verbatim from mod_cpp_fw_v1.4e will have an extension [FW], SFRs copied from mod_vpngw_v1.3 will have extension [VPN] and SFRs copied from mod_ips_v1.0 will have extension [IPS] to distinguish them from the NDcPP. These SFRs that have an extension of [FW], [IPS] or [VPN] do not exist in NDcPPv2.2e. Changes have been made to the base cPP SFRs as necessary to support the Intrusion Prevention, firewall and VPN functionality based on mod_ips_v1.0, mod_cpp_fw_v1.4e and mod_vpngw_v1.3.

Except where noted, all aspects of SFRs are applicable to entire TOE (FTD, FMC and FXOS). Where specific functionality is only implemented in either FTD or FXOS, the applicable subcomponent is identified in an application note, or in embedded qualifiers within the text of the SFR. Application notes clarify distinctions where the TOE includes multiple implementations of a functionality and those implementations differ in their minimum support of the functionality. Thus, the SFR is stating the combined functionality of the TOE.

5.2 TOE Security Functional Requirements

This section identifies the Security Functional Requirements for the TOE. The TOE Security Functional Requirements that appear in the following table are described in more detail in the following subsections.

Table 20: Security Functional Requirements

Class Name	Component Identification	Component Name
Reproduced from cpp_nd_v2.2e		
FAU: Security Audit	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User identity association

Class Name	Component Identification	Component Name
	FAU_GEN_EXT.1	Security Audit Generation
	FAU_STG_EXT.1	Protected Audit Event Storage
	FAU_STG_EXT.4	Protected Local Audit Event Storage for Distributed TOEs
	FAU_STG_EXT.5	Protected Remote Audit Event Storage for Distributed TOEs
FCO: Communication	FCO_CPC_EXT.1	Component Registration Channel Definition
FCS: Cryptographic Support	FCS_CKM.1	Cryptographic Key Generation
	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_HTTPS_EXT.1	HTTPS Protocol
	FCS_IPSEC_EXT.1(1)	IPsec Protocol-FXOS
	FCS_NTP_EXT.1(1)	NTP Protocol-FXOS
	FCS_NTP_EXT.1(2)	NTP Protocol-FMC
	FCS_RBG_EXT.1	Random Bit Generation
	FCS_SSHS_EXT.1(1)	SSH Server Protocol (FXOS)
	FCS_SSHS_EXT.1(2)	SSH Server Protocol (FTD/FMC/FMCv)
	FCS_TLSC_EXT.1	TLS Client Protocol Without Mutual Authentication
	FCS_TLSC_EXT.2	TLS Client Support for Mutual Authentication
	FCS_TLSS_EXT.1	TLS Server Protocol

Class Name	Component Identification	Component Name
	FCS_TLSS_EXT.2	TLS Server Support for Mutual Authentication
FIA: Identification and Authentication	FIA_AFL.1	Authentication Failure Management
	FIA_PMG_EXT.1	Password Management
	FIA_UIA_EXT.1	User Identification and Authentication
	FIA_UAU_EXT.2	Password-based Authentication Mechanism
	FIA_UAU.7	Protected Authentication Feedback
	FIA_X509_EXT.1/ITT	X.509 Certificate Validation
	FIA_X509_EXT.1/Rev	X.509 Certificate Validation
	FIA_X509_EXT.2(1)	X.509 Certificate Authentication [FTD OS TLS client]
	FIA_X509_EXT.2(2)	X.509 Certificate Authentication [FTD TLS Client, FMC and FXOS]
	FIA_X509_EXT.3	X.509 Certificate Requests
FMT: Security Management	FMT_MOF.1/ManualUpdate	Management of Security Functions Behaviour
	FMT_MOF.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_SMR.2	Restrictions on Security Roles
FPT: Protection of the TSF	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_STM_EXT.1	Reliable Time Stamps
	FPT_TST_EXT.1	TSF Testing
	FPT_TUD_EXT.1	Trusted Update

Class Name	Component Identification	Component Name
	FPT_ITT.1	Basic internal TSF data transfer protection
FTA: TOE Access	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: Trusted path/channels	FTP_ITC.1	Inter-TSF Trusted Channel
	FTP_TRP.1/Admin	Trusted Path
Reproduced from mod_ips_v1.0		
FAU: Security Audit	FAU_GEN.1/IPS[IPS]	Audit Data Generation (IPS)
	FAU_SAR.1[IPS]	Audit Review
	FAU_SAR.2[IPS]	Restricted Audit Review
	FAU_SAR.3[IPS]	Selectable Audit Review
	FAU_STG.1/IPS[IPS]	Protected Audit Trail Storage (IPS Data)
FMT: Security Management	FMT_SMF.1/IPS[IPS]	Specification of Management Functions (IPS)
IPS: Intrusion Prevention	IPS_ABD_EXT.1[IPS]	Anomaly-Based IPS Functionality
	IPS_IPB_EXT.1[IPS]	IP Blocking
	IPS_NTA_EXT.1[IPS]	Network Traffic Analysis
	IPS SBD_EXT.1[IPS]	Signature-Based IPS Functionality
Reproduced from mod_cpp_fw_v1.4e		
FDP: User Data Protection	FDP_RIP.2[FW]	Full Residual Information Protection
FFW: Stateful Traffic Filtering	FFW_RUL_EXT.1[FW]	Stateful Traffic Filtering
	FFW_RUL_EXT.2[FW]	Stateful Filtering of Dynamic Protocols
FMT: Security Management	FMT_SMF.1/FFW[FW]	Specification of Management Functions

Class Name	Component Identification	Component Name
Reproduced from mod_vpngw_v1.3		
FAU: Security Audit	FAU_GEN.1/VPN[VPN]	Audit Data Generation (VPN Gateway)
FCS: Cryptographic Support	FCS_CKM.1/IKE[VPN]	Cryptographic Key Generation (for IKE Peer Authentication)
	FCS_IPSEC_EXT.1(2)[VPN]	IPsec Protocol – FTD
FMT: Security Management	FMT_SMF.1/VPN[VPN]	Specification of Management Functions (VPN Gateway)
FPF: Packet Filtering	FPF_RUL_EXT.1[VPN]	Rules for Packet Filtering
FPT: Protection of the TSF	FPT_FLS.1/SelfTest[VPN]	Fail Secure (Self-Test Failures)
	FPT_TST_EXT.3[VPN]	Self-Test with Defined Methods
FTA: TOE Access	FTA_SSL.3/VPN[VPN]	TSF-Initiated Termination (VPN Headend)
	FTA_TSE.1[VPN]	TOE Session Establishment
	FTA_VCM_EXT.1[VPN]	VPN Client Management
FTP: Trusted path/channels	FTP_ITC.1/VPN[VPN]	Inter-TSF Trusted Channel (VPN Communications)

5.3 SFRs Drawn from NDcPP

5.3.1 Security audit (FAU)

5.3.1.1 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 <u>not specified</u> 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).*
- *[<u>no other actions</u>];*

d) *Specifically defined auditable events listed in Table 21.*

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 21*.

Table 21: Auditable Events

SFR	Auditable Event	Additional Audit Record Contents
Reproduced from NDcPP		
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_GEN_EXT.1	None.	None.
FAU_STG.1	None.	None.
FAU_STG_EXT.1	None.	None.
FAU_STG_EXT.4	None.	None.
FAU_STG_EXT.5	None.	None.
FCO_CPC_EXT.1	• Enabling communications between a pair of components. • Disabling communications between a pair of components.	Identities of the endpoints pairs enabled or disabled.
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_HTTPS_EXT.1	Failure to establish a HTTPS session.	Reason for failure
FCS_IPSEC_EXT.1(1)	Failure to establish an IPsec SA.	Reason for failure.
FCS_NTP_EXT.1(1)	• Configuration of a new time server • Removal of configured time server	Identity if new/removed time server
FCS_NTP_EXT.1(2)	• Configuration of a new time server • Removal of configured time server	Identity if new/removed time server
FCS_RBG_EXT.1	None.	None.
FCS_SSHS_EXT.1(1)	Failure to establish an SSH session	Reason for failure

SFR	Auditable Event	Additional Audit Record Contents
FCS_SSHS_EXT.1(2)	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSC_EXT.2	None.	None.
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.2	Failure to authenticate the client	Reason for failure
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 the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/ITT	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.1/Rev	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2(1)	None.	None.
FIA_X509_EXT.2(2)	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Services	None.	None.
FMT_MTD.1/CoreData	None	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).

SFR	Auditable Event	Additional Audit Record Contents
FPT_ITT.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt
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. • Failures of the trusted path functions.	None.
Reproduced from the mod_cpp_fw_v1.4e		
FDP_RIP.2[FW]	None.	None.
FFW_RUL_EXT.1[FW]	Application of rules configured with the ‘log’ operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface
FFW_RUL_EXT.2[FW]	Dynamical definition of rule Establishment of a session	None.
FMT_SMF.1/FFW[FW]	All management activities of TSF data(including creation, modification and deletion of firewall rules).	None.
Reproduced from the mod_vpngw_v1.3		
FAU_GEN.1/VPN[VPN]	No events specified.	N/A
FCS_CKM.1/IKE[VPN]	No events specified.	N/A
FCS_IPSEC_EXT.1(2)[VPN]	Failure to establish an IPsec SA.	Reason for failure.
FMT_SMF.1/VPN[VPN]	All administrative actions	No additional information.

SFR	Auditable Event	Additional Audit Record Contents
FPF_RUL_EXT.1[VPN]	Application of rules configured with the 'log' operation	• Source and destination addresses • Source and destination ports • Transport Layer Protocol
FPT_FLS.1/SelfTest[VPN]	No events specified.	N/A
FPT_TST_EXT.3[VPN]	No events specified.	N/A
FTA_SSL.3/VPN[VPN]	No events specified.	N/A
FTA_TSE.1[VPN]	No events specified.	N/A
FTA_VCM_EXT.1[VPN]	No events specified.	N/A
FTP_ITC.1/VPN[VPN]	Initiation of the trusted channel	No additional information.
	Termination of the trusted channel	No additional information.
	Failure of the trusted channel functions	Identification of the initiator and target of failed trusted channel establishment attempt

5.3.1.2 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.

5.3.1.3 FAU_GEN_EXT.1 Security Audit Generation

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

5.3.1.4 FAU_STG_EXT.1 Protected Audit Event Storage

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

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

- *The TOE shall be a distributed TOE that stores audit data on the following TOE components: [FMC, FTD, FXOS]*,
- *The TOE shall be a distributed TOE with storage of audit data provided externally for the following TOE components: [FTD transmits IPS events audit data to FMC]*.]

FAU_STG_EXT.1.3 The TSF shall [*overwrite previous audit records according to the following rule: [the newest audit record will overwrite the oldest audit record]*] when the local storage space for audit data is full.

5.3.1.5 FAU_STG_EXT.4 Protected Local Audit Event Storage for Distributed TOEs

FAU_STG_EXT.4.1 The TSF of each TOE component which stores security audit data locally shall perform the following actions when the local storage space for audit data is full:

[

FMC: overwrite previous audit records according to the following rule: [oldest records are overwritten]

FXOS: overwrite previous audit records according to the following rule: [oldest records are overwritten]

FTD: overwrite previous audit records according to the following rule: [oldest records are overwritten]

]

5.3.1.6 FAU_STG_EXT.5 Protected Remote Audit Event Storage for Distributed TOEs

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

5.3.2 Communication (FCO)

5.3.2.1 FCO_CPC_EXT.1 Communication Partner Control

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

FCO_CPC_EXT.1.2 The TSF shall implement a registration process in which components establish and use a communications channel that uses [

- *A channel that meets the secure channel requirements in [FPT_ITT.1]*].

for at least TSF data.

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

Application Note

FPT_ITT.1 is also iterated as FPT_ITT.1/Join. The FPT_ITT.1/Join iteration is associated with FCO_CPC_EXT.1.2 for initial registration of the FTD to FMC.

5.3.3 Cryptographic Support (FCS)

5.3.3.1 FCS_CKM.1 Cryptographic Key Generation

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

- *RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;*
- *ECC schemes using "NIST curves" [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4*
- *FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and* [*RFC 3526, RFC 7919*]

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

5.3.3.2 FCS_CKM.2 Cryptographic Key Establishment (Refinement)

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

- *RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1";*
- *Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"*
- *FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and* [*groups listed in RFC 3526*]

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

5.3.3.3 FCS_CKM.4 Cryptographic Key Destruction

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

- *For plaintext keys in volatile storage, the destruction shall be executed by a [single overwrite consisting of [zeroes], destruction of reference to the key directly followed by a request for garbage collection];*
- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [*
  - *logically addresses the storage location of the key and performs a [[one]-pass] overwrite consisting of [zeroes];*
  - *instructs a part of the TSF to destroy the abstraction that represents the key]*

that meets the following: *No Standard*.

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

FCS_COP.1.1/DataEncryption The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in **[*CBC, GCM*] and [*no other*]** mode and cryptographic key sizes **[*128 bits, 256 bits*] and [*192 bits*]** that meet the following: AES as specified in ISO 18033-3, ***[CBC as specified in ISO 10116, GCM as specified in ISO 19772]*** **and [*no other standards*]**.

5.3.3.5 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, 3072 bits]*
- *Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256, 384, and 521 bits]*

]

that meet the following: [

- *For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS2v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,*

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

].

5.3.3.6 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*.

5.3.3.7 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 and 512 bits*] **and message digest sizes *[160, 256, 384, 512] bits*** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

5.3.3.8 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.

5.3.3.9 FCS_IPSEC_EXT.1(1) IPsec Protocol - FXOS

FCS_IPSEC_EXT.1.1(1) The TSF shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2(1) 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(1) The TSF shall implement [*transport mode, tunnel mode*].

FCS_IPSEC_EXT.1.4(1) The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [*AES-CBC-128 (RFC 3602), AES-CBC-192 (RFC 3602), AES-CBC-256 (RFC 3602), AES-GCM-128 (RFC 4106)*] together with a Secure Hash Algorithm (SHA)-based HMAC [*HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512*].

FCS_IPSEC_EXT.1.5(1) The TSF shall implement the protocol: [

- *IKEv2 as defined in RFC 5996 and [with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)], and [RFC 4868 for hash functions]*

].

FCS_IPSEC_EXT.1.6(1) The TSF shall ensure the encrypted payload in the [*IKEv2*] protocol uses the cryptographic algorithms [*AES-CBC-128, AES-CBC-192, AES-CBC-256 (specified in RFC 3602), AES-GCM-128 (specified in RFC 5282)*].

FCS_IPSEC_EXT.1.7(1) The TSF shall ensure that [

- *IKEv2 SA lifetimes can be configured by a Security Administrator based on*

  [

  - *length of time, where the time values can be configured within [60-1440 minutes, 1-24] hours*

  ]

].

FCS_IPSEC_EXT.1.8(1) The TSF shall ensure that [

- *IKEv2 Child SA lifetimes can be configured by a Security Administrator based on*

  [

  - *length of time, where the time values can be configured within [30-480 minutes, 0.5-8] hours;*

  ]

].

FCS_IPSEC_EXT.1.9(1) 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 [*512*] bits.

FCS_IPSEC_EXT.1.10(1) The TSF shall generate nonces used in [*IKEv2*] exchanges of length [

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

] .

FCS_IPSEC_EXT.1.11(1) The TSF shall ensure that IKE protocols implement DH Group(s)

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

].

FCS_IPSEC_EXT.1.12(1) The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [*IKEv2 IKE_SA*] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [*IKEv2 CHILD_SA*] connection.

FCS_IPSEC_EXT.1.13(1) The TSF shall ensure that all IKE protocols perform peer authentication using [*RSA*] that use X.509v3 certificates that conform to RFC 4945 and [*no other method*].

FCS_IPSEC_EXT.1.14(1) 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)*]

5.3.3.10 FCS_NTP_EXT.1(1) NTP Protocol - FXOS

FCS_NTP_EXT.1.1(1) The TSF shall use only the following NTP version(s) [*NTP v3 (RFC 1305)*].

FCS_NTP_EXT.1.2(1) The TSF shall update its system time using [

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

  ].

FCS_NTP_EXT.1.3(1) The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

FCS_NTP_EXT.1.4(1) The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

5.3.3.11 FCS_NTP_EXT.1(2) NTP Protocol - FMC

FCS_NTP_EXT.1.1(2) The TSF shall use only the following NTP version(s) [*NTP v4 (RFC 5905)*].

FCS_NTP_EXT.1.2(2) The TSF shall update its system time using [

- Authentication using [*SHA-1*] as the message digest algorithm(s);

  ].

FCS_NTP_EXT.1.3(2) The TSF shall not update NTP timestamp from broadcast and/or multicast addresses.

FCS_NTP_EXT.1.4(2) The TSF shall support configuration of at least three (3) NTP time sources in the Operational Environment.

5.3.3.12 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 [*HMAC_DRBG(any)*].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [*[one] platform-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.

5.3.3.13 FCS_SSHS_EXT.1(1) SSH Server Protocol (FXOS)

FCS_SSHS_EXT.1.1(1) The TSF shall implement the SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, [5647].

FCS_SSHS_EXT.1.2(1) The TSF shall ensure that the SSH protocol implementation supports the following user authentication methods as described in RFC 4252: public key-based, [*password-based*].

FCS_SSHS_EXT.1.3(1) The TSF shall ensure that, as described in RFC 4253, packets greater than [*262126*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4(1) The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [*aes128-cbc, aes256-cbc, AEAD_AES_256_GCM*].

FCS_SSHS_EXT.1.5(1) The TSF shall ensure that the SSH public-key based authentication implementation uses [*rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp384*] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6(1) The TSF shall ensure that the SSH transport implementation uses *[hmac-sha1, hmac-sha2-256, hmac-sha2-512, AEAD_AES_256_GCM]* as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7(1) The TSF shall ensure that [*diffie-hellman-group14-sha1]* and *[ecdh-sha2-nistp384]* are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8(1) 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.

5.3.3.14 FCS_SSHS_EXT.1(2) SSH Server Protocol (FTD/FMC/FMCv)

FCS_SSHS_EXT.1.1(2) The TSF shall implement the SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, [5647].

FCS_SSHS_EXT.1.2(2) The TSF shall ensure that the SSH protocol implementation supports the following user authentication methods as described in RFC 4252: public key-based, [*password-based*].

FCS_SSHS_EXT.1.3(2) The TSF shall ensure that, as described in RFC 4253, packets greater than [*262126*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4(2) The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [*aes128-cbc, aes256-cbc, AEAD_AES_128_GCM, AEAD_AES_256_GCM*].

FCS_SSHS_EXT.1.5(2) The TSF shall ensure that the SSH public-key based authentication implementation uses [*rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp384*] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6(2) The TSF shall ensure that the SSH transport implementation uses *[hmac-sha1, hmac-sha2-256, hmac-sha2-512, AEAD_AES_128_GCM, AEAD_AES_256_GCM]* as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7(2) The TSF shall ensure that [*diffie-hellman-group14-sha1(FTD-only), ecdh-sha2-nistp256(FTD-only)*] and [*ecdh-sha2-nistp384, ecdh-sha2-nistp521(FTD-only)*] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8(2) 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.

5.3.3.15 FCS_TLSC_EXT.1 TLS Client Protocol Without Mutual Authentication

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

Relevant to syslog over TLS from FTD TLS Client (No mutual authentication):

- *TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289)*

Relevant to FPT_ITT.1 (No mutual authentication)

- *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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*

- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
- *TLS_DHE_RSA_WITH_AES_128_CBC_SHA256* as *defined in RFC 5246*
- *TLS_DHE_RSA_WITH_AES_256_CBC_SHA256* as *defined in RFC 5246*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*
- *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_256_CBC_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*

Relevant to syslog over TLS from FTD OS TLS Client (No Mutual authentication):

- *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_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*

Relevant to syslog over TLS from FMC/FMCv (With Mutual authentication):

- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246 (TLSv1.2, TLSv1.1)*
- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246 (TLSv1.2, TLSv1.1)*

- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289 (TLSv1.2 only)*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289 (TLSv1.2 only)*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289 (TLSv1.2, TLSv1.1)*

] and no other ciphersuites.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches [*the reference identifier per RFC 6125 section 6, the identifier per RFC 5280 Appendix A using [id-at-title] and no other attribute types*]

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

- *Not implement any administrator override mechanism*.

]

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

Application Note

FCS_TLSC_EXT.1 is applicable to two TLS clients that send syslog messages to the syslog server- ***FTD TLS client,*** *that is configured by the FMC and is the main audit system for audits generated by FTD. It sends audit events such as IPsec and login messages to the external syslog server and Mutual authentication is not supported; and the* ***FTD OS TLS client****, that is configured through the FTD's command line and sends audit events to an external syslog server such as SSH login, console login, etc. and Mutual authentication is not supported.*

The selection of – "the identifier per RFC 5280 Appendix A using [id-at-title]"in FCS_TLSC_EXT.1.2 is only applicable to the TLS connection that is relevant to FPT_ITT.1

5.3.3.16 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

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

Application Note

FCS_TLSC_EXT.2 is applicable to the TLS client in FMC/FMCv that is used for transmission of syslog over TLS.

5.3.3.17 FCS_TLSS_EXT.1 TLS Server Protocol

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

Relevant to FPT_ITT.1 (FMC as server and FTD as client):

- *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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*

Relevant to FTP_TRP.1/Admin (applicable to FMC/FMCv only):

- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*

Relevant to FTP_TRP.1/Admin (applicable to FXOS only):

- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246*
- *TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288*
- *TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
- *TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289*

] and no other ciphersuites.

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [*TLS 1.1(FMC/FMCv-only)*].

FCS_TLSS_EXT.1.3 The TSF shall perform key establishment for TLS using [*RSA with key size [2048 bits], ECDHE curves [secp256r1, secp384r1, secp521r1] and no other curves*].

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

5.3.3.18 FCS_TLSS_EXT.2 TLS Server Support for Mutual Authentication

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

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

- *Not implement any administrator override mechanism*

].

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

Application Note

This SFR only applies to the FPT_ITT.1 channel for the FMC/FMCv side (FMC/FMCv as server and FTD is the client)

5.3.4 Identification and authentication (FIA)

5.3.4.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1 to 999 **(FMC)**, 1 to 10 **(FXOS)**, 1 to 9999 **(FTD)***] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

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

5.3.4.2 FIA_PMG_EXT.1 Password Management

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, [“ ” ‘ ` (double or single quote/apostrophe), + (plus), - (minus), = (equal), , (comma), . (period), / (forward-slash), \ (back-slash), | (vertical-bar or pipe), : (colon), ; (semi-colon), < > (less-than, greater-than inequality signs), [ ] (square-brackets), { } (braces or curly-brackets),^ (caret), _ (underscore), and ~ (tilde)]];

b) Minimum password length shall be configurable to between *[1(FTD), 8(FMC, FXOS)] and [127]* characters.

5.3.4.3 FIA_UIA_EXT.1 User Identification and Authentication

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

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

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

5.3.4.4 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local [*password-based, [support for RADIUS and TACACS+](FXOS-only)*] authentication mechanism to perform local administrative user authentication.

5.3.4.5 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**.

5.3.4.6 FIA_X509_EXT.1/ITT X.509 Certificate Validation

FIA_X509_EXT.1.1/ITT 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 two certificates.
- The certificate path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [*no revocation method*].

- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

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

5.3.4.7 FIA_X509_EXT.1/Rev X.509 Certificate Validation

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

- RFC 5280 certificate validation and certificate path validation **supporting a minimum path length of three certificates**.
- The certificate path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [*the Online Certificate Status Protocol (OCSP) as specified in RFC 6960* ***(FTD-only)****, a Certificate Revocation List (CRL) as specified in RFC 5759 Section 5*].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
  - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
  - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
  - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

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

5.3.4.8 FIA_X509_EXT.2(1) X.509 Certificate Authentication [FTD OS TLS Client]

FIA_X509_EXT.2.1(1) The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [*TLS*], and [*no additional uses*].

FIA_X509_EXT.2.2(1) When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [*not accept the certificate*].

5.3.4.9 FIA_X509_EXT.2(2) X.509 Certificate Authentication [FTD TLS Client, FMC and FXOS]

FIA_X509_EXT.2.1(2) The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for **IPsec and** [***TLS***], and [*no additional uses*].

FIA_X509_EXT.2.2(2) When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall [*not accept the certificate*].

5.3.4.10 FIA_X509_EXT.3 X.509 Certificate Requests

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

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

5.3.5 Security management (FMT)

5.3.5.1 FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

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

5.3.5.2 FMT_MOF.1/Services Management of Security Functions Behaviour

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

5.3.5.3 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*.

5.3.5.4 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*].

5.3.5.5 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] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- [
    - *Ability to start and stop services;*
    - *Ability to modify the behavior of the transmission of audit data to an external IT entity;*
    - *Ability to manage the cryptographic keys;*
    - *Ability to configure the cryptographic functionality;*
    - *Ability to configure thresholds for SSH rekeying;*
    - *Ability to configure the lifetime for IPsec SAs;*
    - *Ability to configure the interaction between TOE components;*
    - *Ability to configure NTP;*
    - *Ability to manage the TOE's trust store and designate X509.v3 certificates as trust*
    - *anchors;*
    - *Ability to manage the trusted public keys database;*
    - *Ability to re-enable an Administrator account;*
    - *Ability to set the time which is used for time-stamps;*
    - *Ability to configure the reference identifier for the peer;*
    - *Ability to import X.509v3 certificates to the TOE's trust store;*

    ]

5.3.5.6 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)

5.3.6.1 FPT_SKP_EXT.1 Protection of TSF Data (for Reading of All Symmetric Keys)

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

5.3.6.2 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.

5.3.6.3 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 **(FMC and FXOS)**, synchronise time with an NTP server **(FXOS and FMC)***].

5.3.6.4 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: **noise source health tests**, [*FIPS 140-2 standard power-up self-tests and firmware integrity test*].

5.3.6.5 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1.1 The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software and [*the most recently installed version of the TOE firmware/software*].

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.

5.3.6.6 FPT_ITT.1: Basic Internal TOE TSF data transfer protection

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

5.3.7 TOE Access (FTA)

5.3.7.1 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.

5.3.7.2 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.*

5.3.7.3 FTA_SSL.4 User-initiated Termination

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

5.3.7.4 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/Channels (FTP)

5.3.8.1 FTP_ITC.1 Inter-TSF Trusted Channel

FTP_ITC.1.1 The TSF shall **be capable of using [*IPsec, TLS*] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [*VPN Communications, NTP server, Authentication server (FXOS only)*]** 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: transmit audit data via syslog over IPsec **(FTD and FXOS)** or TLS **(FMC and FTD)**;*
- *NTP server using IPsec **(FXOS only)**;*
- *Authentication servers (RADIUS, TACAC+) using IPsec **(FXOS only)***].

5.3.8.2 FTP_TRP.1/Admin Trusted Path

FTP_TRP.1.1/Admin The TSF shall **be capable of using** [***SSH, TLS (FMC and FXOS only), HTTPS (FMC and FXOS only), IPsec (FTD and FXOS only)***] **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 communicated 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*.

5.4 SFRs Drawn from mod_ips_v1.0

5.4.1 Security Audit (FAU)

5.4.1.1 FAU_GEN.1/IPS[IPS] Audit Data Generation (IPS)

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

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

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

c) [*All dissimilar IPS events;*

d) All dissimilar IPS reactions;

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

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

g) The events in the IPS Events table.

h) [no other auditable events]

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

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

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

Table 22: Auditable Events

SFR	Auditable Event	Additional Audit Record Contents
FMT_SMF.1/IPS[IPS]	Modification of an IPS policy element.	Identifier or name of the modified IPS policy element (e.g. which signature or known-good/known-bad list was modified).
IPS_ABD_EXT.1[IPS]	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[IPS]	Inspected traffic matches a list of known-good or known-bad addresses applied to an IPS policy.	Source and destination IP addresses (and, if applicable, indication of whether the source and/or destination address matched the list).
		TOE interface that received the packet.
		Network-based action by the TOE (e.g. allowed, blocked, sent reset).
IPS_NTA_EXT.1[IPS]	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[IPS]	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).

5.4.1.2 FAU_SAR.1[IPS] Audit Review

FAU_SAR.1.1[IPS] Refinement: The TSF shall provide [*authorized administrators*] with the capability to read [*IPS data*] from the ~~audit records~~ **IPS events**.

FAU_SAR.1.2[IPS] Refinement: The TSF shall provide the ~~audit records~~ **IPS data** in a manner suitable for the ~~user~~ **administrators** to interpret the information.

5.4.1.3 FAU_SAR.2[IPS] Restricted Audit Review

FAU_SAR.2.1[IPS] The TSF shall prohibit all ~~users~~ **administrators** read access to the ~~audit records~~ **IPS data,** except those that have been granted explicit read-access.

5.4.1.4 FAU_SAR.3[IPS] Selectable Audit Review

FAU_SAR.3.1[IPS] The TSF shall provide the ability to apply [*filtering and sorting*] of ~~audit~~ **IPS data** based on [*filtering parameters: risk rating, time period, source IP address, destination IP address and* [*<u>other filtering parameters described in the TSS</u>*]; *and sorting parameters: event ID, event type, time, signature ID, IPS actions performed, and* [[*<u>other sorting parameters described in the TSS</u>*]].

5.4.1.5 FAU_STG.1/IPS[IPS] Protected Audit Trail Storage (IPS Data)

FAU_STG.1.1/IPS[IPS] The TSF shall protect the stored ~~audit records~~ **IPS data** from unauthorized deletion.

FAU_STG.1.2/IPS[IPS] The TSF shall be able to [*prevent*] unauthorized modifications to the stored ~~audit records~~ **IPS data** ~~in the audit trail~~.

5.4.2 Security Management (FMT)

5.4.2.1 FMT_SMF.1/IPS[IPS] Specification of Management Functions (IPS)

FMT_SMF.1.1/IPS[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*]

5.4.3 Intrusion Prevention (IPS)

5.4.3.1 IPS_ABD_EXT.1[IPS] Anomaly-Based IPS Functionality

IPS_ABD_EXT.1.1[IPS] The TSF shall support the definition of [*anomaly ('unexpected') traffic patterns*] including the specification of [

- *frequency;*
- *[preprocessor detection rules for anomaly detected in headers and protocols]*]

and the following network protocol fields:

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

Application Note

Although the term "threshold" is used in the TSS, the TOE's definition of "threshold" matches the definition of frequency in the mod_ips_v1.0. Therefore, "frequency", rather than "threshold" has been selected in the IPS_ABD_EXT.1.1 requirement.

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

IPS_ABD_EXT.1.3[IPS] 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]*

5.4.3.2 IPS_IPB_EXT.1[IPS] IP Blocking

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

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

5.4.3.3 IPS_NTA_EXT.1[IPS] Network Traffic Analysis

IPS_NTA_EXT.1.1[IPS] 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[IPS] 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[IPS] 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: [*Giga Ethernet*];
- Inline (data pass-through) mode: [*Giga Ethernet*];

- Management mode: [*Giga Ethernet*];
- [
  - o *no other interface types*].

5.4.3.4 IPS_SBD_EXT.1[IPS] Signature-Based IPS Functionality

IPS_SBD_EXT.1.1[IPS] 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 [no other field].*
- *ICMP: Type; Code; Header Checksum; and [ID, sequence number, [no other field]].*
- *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[IPS] 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:*
  i) 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.

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

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

  iv) [no other types of TCP payload inspection];
- *UDP data: characters beyond the first 8 bytes of the UDP header;*
- *[no other types of packet payload inspection]*

IPS_SBD_EXT.1.3[IPS] 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:

a) IP Attacks

i) IP Fragments Overlap (Teardrop attack, Bonk attack, or Boink attack)

ii) IP source address equal to the IP destination (Land attack)

b) ICMP Attacks

i) Fragmented ICMP Traffic (e.g. Nuke attack) *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

iv) TCP SYN+RST flags

d) UDP Attacks

i) UDP Bomb Attack

ii) UDP Chargen DDoS Attack

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

a) Flooding a host (DoS attack)

i) ICMP flooding (Smurf attack, and ping flood)

ii) TCP flooding (e.g. SYN flood)

b) Flooding a network (DoS attack)

c) Protocol and port scanning

i) IP protocol scanning

ii) TCP port scanning

iii) UDP port scanning

iv) ICMP scanning

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

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

  ]

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

5.5 SFRs Drawn from mod_cpp_fw_v1.4e

5.5.1 User Data Protection (FDP)

5.5.1.1 FDP_RIP.2[FW] Full Residual Information Protection

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

5.5.2 Stateful Traffic Filtering (FFW)

5.5.2.1 FFW_RUL_EXT.1[FW] Stateful Traffic Filtering

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

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

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

and distinct interface.

FFW_RUL_EXT.1.3[FW] 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[FW] The TSF shall allow the stateful traffic filtering rules to be assigned to each distinct network interface.

FFW_RUL_EXT.1.5[FW] 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, [no other protocols] 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. [no other protocols].

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[FW] The TSF shall enforce the following default stateful traffic filtering rules on all network traffic:

a) The TSF shall drop and be capable of [counting] packets which are invalid fragments;
b) The TSF shall drop and be capable of [counting] fragmented packets which cannot be re-assembled completely;
c) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a broadcast network;
d) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a multicast network;
e) The TSF shall drop and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
f) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address "reserved for future use" (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
g) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;

h) The TSF shall drop and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
i) [[Other traffic dropped by default and able to be logged:

i. Slowpath Security Checks – The TSF shall reject and be capable of logging the detection of the following network packets:
1. In routed mode when the TOE receives a through-the-box:
a. L2 broadcast packet (MAC address FF:FF:FF:FF:FF:FF)
b. IPv4 packet with destination IP address equal to 0.0.0.0
c. IPv4 packet with source IP address equal to 0.0.0.0
2. In routed or transparent mode when the TOE receives a through-the-box IPv4 packet with any of:
a. first octet of the source IP address equal to zero

b. *network part of the source IP address equal to all 0's*
c. *network part of the source IP address equal to all 1's*
d. *source IP address host part equal to all 0's or all 1's*
e. *source IP address and destination IP address are the same (“land.c” attack)*

ii. *LAND Attack: The TSF shall reject and be capable of logging network packets with the IP source address equal to the IP destination, and the destination port equal to the source port.*

iii. *ICMP Error Inspect and ICMPv6 Error Inspect - The TSF shall reject and be capable of logging ICMP error packets when the ICMP error messages are not related to any session already established in the TOE.*

iv. *ICMPv6 condition - The TSF shall reject and be capable of logging network packets when the appliance is not able to find any established connection related to the frame embedded in the ICMPv6 error message.*

v. *ICMP Inspect bad icmp code - The TSF shall reject and be capable of logging network packets when an ICMP echo request/reply packet was received with a malformed code(non-zero)]].*

FFW_RUL_EXT.1.7[FW] 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[FW] The TSF shall process the applicable stateful traffic filtering rules in an administratively defined order.

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

FFW_RUL_EXT.1.10[FW] 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 [counted, logged].*

5.5.2.2 FFW_RUL_EXT.2[FW] Stateful Filtering of Dynamic Protocols

FFW_RUL_EXT.2.1[FW] The TSF shall dynamically define rules or establish sessions allowing network traffic to flow for the following network protocols [*FTP*].

5.5.3 Security Management (FMT)

5.5.3.1 FMT_SMF.1/FFW[FW] Specification of Management Functions

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

- *Ability to configure firewall rules*

5.6 SFRs from mod_vpngw_v1.3

5.6.1 Security audit (FAU)

5.6.1.1 FAU_GEN.1/VPN[VPN] Audit Data Generation (VPN Gateway)

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

a. Start-up and shutdown of the audit functions

b. Indication that TSF self-test was completed

c. Failure of self-test

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

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

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

a. Date and time of the event, type of event, subject identity (if applicable), and the outcome (success or failure) of the event; and
b. For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, [*additional information*
c. *defined in the Auditable Events for Mandatory Requirements table for each auditable event, where applicable].*

5.6.2 Cryptographic Support (FCS)

5.6.2.1 FCS_CKM.1/IKE[VPN] Cryptographic Key Generation (for IKE Peer Authentication)

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

- FIPS PUB 186-5, “Digital Signature Standard (DSS)”; Appendix B.3 for RSA schemes;
- FIPS PUB 186-5, “Digital Signature Standard (DSS)”, Appendix B.4 for ECDSA schemes, and implementing “NIST curves” P-384 and [P-256, P-521]

]

and

[no other key generation algorithm]

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

5.6.2.2 FCS_IPSEC_EXT.1(2)[VPN] IPsec Protocol-FTD

FCS_IPSEC_EXT.1.1(2)[VPN] The TSF shall implement the IPsec architecture as specified in RFC 4301.

FCS_IPSEC_EXT.1.2(2)[VPN] 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(2)[VPN] The TSF shall implement [*transport mode, tunnel mode*].

FCS_IPSEC_EXT.1.4(2)[VPN] The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms **[*AES-CBC-128, AES-CBC-256 (specified in RFC 3602), AES-GCM-128), AES-GCM-256 (specified in RFC 4106)*] and [*no other algorithm*]** together with a Secure Hash Algorithm (SHA)-based HMAC [*HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512*]

FCS_IPSEC_EXT.1.5(2)[VPN] The TSF shall implement the protocol: [

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

].

FCS_IPSEC_EXT.1.6(2)[VPN] The TSF shall ensure the encrypted payload in the [*IKEv2*] protocol uses the cryptographic algorithms [*AES-CBC-128, AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES-GCM-256 (specified in RFC 5282)*].

FCS_IPSEC_EXT.1.7(2)[VPN] The TSF shall ensure that [

- *IKEv2 SA lifetimes can be configured by a Security Administrator based on*

  [

  - *length of time, where the time values can be configured within [120 to 2,147,483,647 seconds. The default is 86,400 seconds or 24] hours*

  ]

].

FCS_IPSEC_EXT.1.8(2)[VPN] The TSF shall ensure that [

- *IKEv2 Child SA lifetimes can be configured by a Security Administrator based on*

  [

  - *number of bytes;*
  - *length of time, where the time values can be configured within [120-2,147,483,647 seconds with the default being 28,800 seconds which is 8] hours;*

]

].

FCS_IPSEC_EXT.1.9(2)[VPN] 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 [*512*] bits.

FCS_IPSEC_EXT.1.10(2)[VPN] The TSF shall generate nonces used in [*IKEv2*] exchanges of length [

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

] .

FCS_IPSEC_EXT.1.11(2)[VPN] The TSF shall ensure that all IKE protocols implement DH Group(s)

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

  ]

FCS_IPSEC_EXT.1.12(2)[VPN] The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [*IKEv2 IKE SA*] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [*IKEv2 CHILD SA*] connection.

FCS_IPSEC_EXT.1.13(2)[VPN] The TSF shall ensure that **[*IKEv2*]** protocols perform peer authentication using [*RSA, ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [*no other method*].

FCS_IPSEC_EXT.1.14(2)[VPN] The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: **Distinguished Name (DN),** [*SAN: Fully Qualified Domain Name (FQDN)*].

Application Note

In FCS_IPSEC_EXT.1.7(2)[VPN], IKEv2 SA can be limited by time only. IKEv2 Child SA can be limited by time or number of kilobytes. The time is in number of seconds.

5.6.3 Security Management (FMT)

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

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

- *Definition of packet filtering rules;*
- *Association of packet filtering rules to network interfaces;*
- *Ordering of packet filtering rules by priority;*

[

- *Configuration of remote VPN client session timeout*
- *Configuration of attributes used to deny establishment of remote VPN client sessions*
- *Generation of bit-based pre-shared key,*

]]

5.6.4 Packet Filtering (FPF)

5.6.4.1 FPF_RUL_EXT.1[VPN] Packet Filtering

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

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

- *IPv4 (RFC 791)*
  - *source address*
  - *destination Address*
  - *protocol*
- *IPv6 (RFC 8200)*
  - *source address*
  - *destination Address*
  - *next header (protocol)*
- *TCP (RFC 793)*
  - *source port*
  - *destination port*
- *UDP (RFC 768)*
  - *source port*
  - *destination port*

FPF_RUL_EXT.1.3[VPN] The TSF shall allow the following operations to be associated with packet filtering rules: permit and drop with the capability to log the operation.

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

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

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

5.6.5 Protection of the TSF (FPT)

5.6.5.1 FPT_FLS.1/SelfTest[VPN] Fail Secure

FPT_FLS.1.1/SelfTest[VPN] 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*]

5.6.5.2 FPT_TST_EXT.3[VPN]: Self-Test with Defined Methods

FPT_TST_EXT.3.1[VPN] The TSF shall run a suite of the following self-tests [*[when loaded for execution]*] to demonstrate the correct operation of the TSF: [*integrity verification of stored executable code*].

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

5.6.6 TOE Access (FTA)

5.6.6.1 FTA_SSL.3/VPN[VPN] TSF-initiated Termination (VPN Headend)

FTA_SSL.3.1/VPN[VPN] The TSF shall terminate **a remote VPN client** session after an [*Administrator-configurable time interval of session inactivity*].

5.6.6.2 FTA_TSE.1[VPN] TOE Session Establishment

FTA_TSE.1.1[VPN] The TSF shall be able to deny session establishment of a **remote VPN client** session based on [*location, time, day,* [***no other attributes***]].

5.6.6.3 FTA_VCM_EXT.1[VPN] VPN Client Management

FTA_VCM_EXT.1.1[VPN] The TSF shall assign a private IP address to a VPN client upon successful establishment of a security session.

5.6.7 Trusted Path/Channels (FTP)

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

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

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

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

5.7 TOE SFR Dependencies Rationale for SFRs Found in NDcPP and PP-modules

The NDcPP and PP modules contain all the requirements claimed in this Security Target. As such the dependencies are not applicable since the PP itself has been approved.

5.8 Security Assurance Requirements

5.8.1 SAR Requirements

The TOE assurance requirements for this ST are taken directly from the NDcPP which are derived from Common Criteria Version 3.1, Revision 5. The assurance requirements are summarized in the table below.

Table 23: Assurance Measures

Assurance Class	Components	Components Description
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	Labeling of the TOE
	ALC_CMS.1	TOE CM Coverage
TESTS	ATE_IND.1	Independent Testing - Conformance
VULNERABILITY ASSESSMENT	AVA_VAN.1	Vulnerability Analysis

5.8.2 Security Assurance Requirements Rationale

This Security Target claims conformance to the NDcPP. This target was chosen to ensure that the TOE has a basic to moderate level of assurance in enforcing its security functions when instantiated in its intended environment which imposes no restrictions on assumed activity on applicable networks. The ST also claims conformance to mod_ips_v1.0, mod_cpp_fw_v1.4e and mod_vpngw_v1.3, which includes refinements to assurance measures for the SFRs defined in the two aforementioned modules including augmenting the vulnerability analysis (AVA_VAN.1) with specific vulnerability testing.

5.9 Assurance Measures

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by Cisco to satisfy the assurance requirements. The table below lists the details.

Table 24: Assurance Measures

Component	How requirement will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.

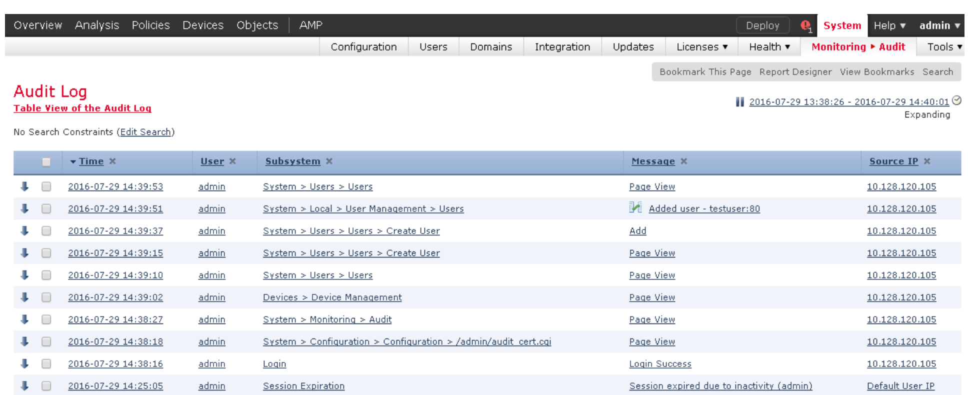
Component	How requirement will be met
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1	The Configuration Management (CM) document(s) describes how the consumer (end-user) of the TOE can identify the evaluated TOE (Target of Evaluation). The CM document(s), identifies the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked, how potential changes are incorporated, and the degree to which automation is used to reduce the scope for error.
ALC_CMS.1	
ATE_IND.1	Cisco provides the TOE for testing.
AVA_VAN.1	Cisco provides the TOE for testing.

6 TOE SUMMARY SPECIFICATION

6.1 TOE Security Functional Requirement Measures

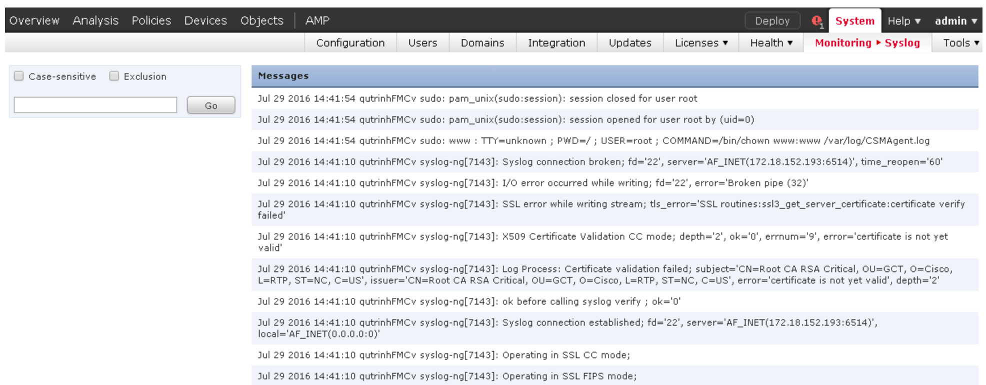
This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

Table 25: How TOE SFRs Are Satisfied

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td colspan="2">Security Functional Requirements Drawn from NDcPP</td></tr>
<tr><td>FAU_GEN.1,
FAU_GEN.1/VPN[VPN],
FAU_GEN_EXT.1,
FAU_STG_EXT.1,
FAU_STG_EXT.4,
FAU_STG_EXT.5</td><td>
Auditing is the recording of events within the system. The TOE generates log records for a wide range of security relevant and other events as they occur. The events that can cause an audit record to be logged include starting the audit function[3], any use of an administrator command or action via the CLI and web interfaces, and all of the required auditable events identified in Table 21. For more information about the required audit events, please refer to Table 21 and the operational user guide (also known as the CC Supplemental User guide).
The FMC component of the TOE can generate an audit record for each user interaction with the web interface and each command in the CLI interface. The FXOS on Firepower 4100/9300 also generates audit records for administrative actions via its CLI (console or SSH), and GUI. The FTD component of the TOE can generate traffic events as part of the access control (firewall), IPS and VPN policies and these event records are stored in logs separate from the audit logs for performance and security reasons. For more details about which auditable events are mapped to which SFRs, refer to Table 34.
FMC and FTD log auditing information for all user activity in a read-only format. Modifications are not allowed by the interfaces and only authorized administrators can delete the audit logs. Audit logs are presented in a standard event view that allows administrators to view, sort, and filter audit log messages based on any item in the audit view. The audit view contains columns with information field for each audit event such as time, user, subsystem, message, and source IP. Please see the figure below for example.
Figure 5: Audit View


</td></tr>
</table>

[3] Note that the audit function cannot be disabled other than shutting down the entire system.

TOE SFRs	How the SFR is Satisfied
	The following fields are recorded for each audit event in the audit view: • **Time**: The time and date that the appliance generated the audit record. • **User**: The user name of the user that triggered the audit event. • **Subsystem**: The menu path the user followed to generate the audit record. For example, "System > Monitoring > Audit" is the menu path to view the audit log. • **Message**: The action the user performed. For example, "Page View" signifies that the user simply viewed the page indicated in the Subsystem, while "Save" means that the user clicked the Save button on the page. • **Source IP**: The IP address of the host used by the user. Figure 6: Syslog View  The user can also view the audit log using the command "show audit-log" or "show syslog" via the CLI interface. All GUI actions and CLI commands are recorded in the audit log and can only be viewed by authorized administrators. To distinguish between the two, the Subsystem field will identify "Command Line" for commands and the Message field will identify the executed command. In general, the logged audit records identify the date and time, the identity of the actor (e.g., user, daemon, or network host) responsible for the event, the subsystem that triggers the event, an indication of whether the event succeeded, failed or had some other outcome (if applicable), and the source IP (if applicable). The logged audit records also include event-specific content that includes at least all of the content required in table above. The TOE (FMC) includes an internal log database implementation that can be used to store and review audit records locally. However, the internal log only stores a default of 100,000 entries in the local database (to configure the size, go to System > Configuration > Database, and click on "Audit Event Database"). When the audit log is full, the oldest audit records are overwritten by the newest audit records. In addition, the TOE (FMC) also includes a local syslog storage in /var/log/messages and these logs are viewable through the FMC GUI. The contents are stored in flat files which are rotated automatically. Similar to the audit log, when the syslog is full, the oldest syslog messages are overwritten by the newest one. For audit log, the events are stored in partitioned event tables. The TOE will prune (i.e., delete) the oldest partition whenever the oldest partition can be pruned without dropping the number of events count below the configured event limit. Note this limit defaults to

TOE SFRs	How the SFR is Satisfied
	10,000 if you set it any lower. For example, if you set the limit to 10,000 events, the events count may need to exceed 15,000 events before the oldest partition can be deleted. For syslog, the logs are stored in /var/log/messages and are rotated daily or when the log file size exceeds 25 MB. After the maximum number of backlog files is reached, the oldest is deleted and the numbers on the other backlogs file are incremented. To prevent the losing of critical audit records, the administrators can configure the system to transmit all the audit events (i.e., audit log and syslog) in real-time over a secure TLS connection or an IPsec connection (FXOS-only) to an external audit server in the operational environment. When an audit event is generated, it is sent to the local storage and external audit server simultaneously. This ensures that current audit events can be viewed locally while all events, new or old, are stored off-line as required by the NDcPP. Note that the protection of the audit records stored at the external audit server is the responsibility of the operational environment. The TOE is only responsible for the secure communication channel. It is recommended that the audit server is physically or logically separated (e.g., VLANs) from the other networks. The TOE can be configured to export syslog records to an administrator-specified, external syslog server. The TOE can be configured to encrypt the communications with an external syslog server using IPsec or TLS. FMC transmits syslog over TLS, FTD transmits syslog over TLS, and FXOS transmits syslog over IPsec. The audit records are also stored locally and when the local storage is full, the newest data will overwrite the oldest data. On FMC, log messages (those generated locally and those forwarded from FTD) are stored locally on FMC in a database. Different message types are stored separately in local databases, and each local store has a separately configurable size limit (configurable in FMC via System > Configuration > Database). Audit events recording FMC administrator actions are stored in the Audit Event Database, network traffic events transmitted from FTD to FMC are stored in separate databases on FMC: firewall events (triggered by Access Control Policy rules) are stored in Connection Database; VPN events are stored in the VPN Troubleshooting Database; and the IPS events are stored in Intrusion Event Database. Messages generated by FTD, including FTD system messages, firewall events, and VPN events are stored locally on FTD and are immediately transmitted from FTD to an external syslog server. As mentioned in the preceding paragraph, the firewall, VPN and IPS events are directly sent to FMC for retention in the FMC databases via secure TLS channel (Note: The IPS events are not stored locally on FTD but are transmitted to an external syslog server via the FMC. IPS events generated on FTD are temporarily stored locally on FTD in a database prior to transmission to FMC). If the connection between FTD and FMC is interrupted, the IPS messages are transmitted once connectivity is restored. As the system, firewall event and VPN event messages are generated by FTD, they are immediately transmitted from FTD to a remote syslog server and stored in a local buffer (buffer size configurable from 4096-52428800 bytes) which overwrites old messages with new ones when storage limits are reached. The local logs are viewable from the FTD CLI by using “show logging”. The local storage of audit events in FXOS (e.g., admin authentication, TLS/HTTPS session state, etc.) is viewable from the “fxos” shell (after using “connect fxos”) by using “show logging”. The storage limit of the local buffer is configurable via FXOS CLI with configurable size limit of 4096-4194304 bytes. This is a circular log (oldest records will be overwritten by new ones when the size limit is reached).

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td></td><td>
For audit messages related to management of cryptographic keys, the audit message details include the name of the certificate associated with the key.
Network interfaces have bandwidth limitations, and other traffic flow limitations that are configurable. When an interface has exceeded a limit for processing traffic, traffic will be dropped, and audit messages can be generated.
For more information on the required auditable events and the actual logs themselves, please refer to the Preparative Procedures & Operational User Guide for the Common Criteria Certified Configuration.
The following high-level events are auditable by the TOE:
<table>
<tr><th>Auditable Event</th><th>Rationale</th></tr>
<tr><td>Modifications to the group of users that are part of the authorized administrator role.</td><td>All changes to the configuration (and hence all security relevant administrator actions) are logged when the logging level is set to at least the 'notifications' level. These changes would fall into the category of configuration changes such as enabling or disabling features and services. The identity of the administrator taking the action and the user being affected (assigned to the authorized administrator role) are both included within the event.</td></tr>
<tr><td>All use of the user identification mechanism.</td><td>Events will be generated for attempted identification/ authentication, and the username attempting to authenticate will be recorded in the event.</td></tr>
<tr><td>Any use of the authentication mechanism.</td><td>Events will be generated for attempted identification/ authentication, and the username attempting to authenticate will be recorded in the event along with the origin or source of the attempt.</td></tr>
<tr><td>The reaching of the threshold for unsuccessful authentication attempts and the subsequent restoration by the authorized administrator of the user's capability to authenticate.</td><td>Failed attempts for authentication will be logged, and when the threshold is reached, it will also be logged.
All changes to the configuration are logged when the logging level is set to at least the 'notifications' level. Changes to restore a locked account would fall into the category of configuration changes.</td></tr>
<tr><td>All decisions on requests for information flow.</td><td>In order for events to be logged for information flow requests, the 'log' keyword may need to be in each line of an access control list. The presumed addresses of the source and destination subjects are included in the event.</td></tr>
<tr><td>Success and failure, and the type of cryptographic operation</td><td>Attempts for VPN connections are logged (whether successful or failed). Requests for encrypted session negotiation are logged</td></tr>
</table>
</td></tr>
</table>

TOE SFRs	How the SFR is Satisfied	
		(whether successful or failed). The identity of the user performing the cryptographic operation is included in the event.
	Failure to establish and/or establishment/termination of an IPsec session	Attempts to establish an IPsec tunnel or the failure of an established IPsec tunnel is logged as well as successfully established and terminated IPsec sessions with peer.
	Establishing session with CA and IPsec peer	The connection to CA's or any other entity (e.g., CDP) for the purpose of certificate verification or revocation check is logged. In addition, the TOE can be configured to capture the packets' contents during the session establishment.
	Changes to the time.	Changes to the time are logged with old and new time values.
	Use of the functions listed in any SFR pertaining to audit.	All changes to the configuration are logged when the logging level is set to at least the 'notifications' level. These changes would fall into the category of configuration changes.
	Loss of connectivity with an external syslog server.	Loss of connectivity with an external syslog server is logged as a terminated or failed cryptographic channel.
	Initiation of an update to the TOE.	TOE updates are logged as configuration changes.
	Termination of local and remote sessions. Note that the TOE does not support session locking, so there is no corresponding audit.	Termination of a local and remote session is logged. This also includes termination of remote VPN session as well. The user may initiate or the system may terminate the session based idle timeout setting.
	Initiation, termination and failures in trusted channels and paths.	Requests for encrypted session negotiation are logged (whether successful or failed). Similarly, when an established cryptographic channel or path is terminated or fails a log record is generated. This applies to HTTPS, TLS, IPsec, and SSH.
	Successful SSH rekey	SSH rekey event is logged.
	Application of rules configured with the 'log' operation	Logs are generated when traffic matches ACLs that are configured with the log operation.

<table>
<tr><th>TOE SFRs</th><th colspan="2">How the SFR is Satisfied</th></tr>
<tr><td></td><td>Indication of packets dropped due to too much network traffic</td><td>Logs are generated when traffic that exceeds the settings allowed on an interface is received.</td></tr>
<tr><td>FAU_GEN.2</td><td colspan="2">The TOE ensures each action performed by the administrator at the CLI and web GUI is logged with the administrator’s identity and as a result events are traceable to a specific user.</td></tr>
<tr><td>FCO_CPC_EXT.1</td><td colspan="2"><u>FTD and FMC</u>

In order for TOE components to communicate as part of a distributed TOE System, they must successfully complete a registration process. Each TOE component comes with a manufacture’s TLS certificate. To start the registration process, the administrator must enable or register the TOE components. On the FMC, the administrator must go to Device Management UI and click on “Add Device”. At the same time, the administrator must go to the FTD CLI, and click or enter “Configure Manager Add”. The administrator must specify the peer hostname or IP address and the registration key used for the initial authentication. During the registration process, the manufacture’s TLS certificates are used to setup the initial TLS channel on the internal trusted management network. If the authentication succeeded, the resident CA on the FMC will sign and issue a TLS certificate along with the private key to the FTD which will be used for subsequent TLS channel. To disable or de-register FTD, the administrator must initiate a “Delete Device” on the FMC Device Management UI and then perform a “Configure Manager Delete” action on the CLI of the FTD. This will destroy (i.e., zeroize) the TLS certificate and private key. Once this has occurred, no farther communication can happen without another registration process.</td></tr>
<tr><td>FCS_CKM.1, FCS_CKM.2, FCS_CKM.4, FCS_COP.1/ DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, FCS_COP.1/ KeyedHash, and FCS_RBG_EXT.1</td><td colspan="2"><u>FTD and FMC</u>

Each FMC and each FTD appliance utilize a cryptographic module (i.e., Cisco FIPS Object Module) to provide supporting cryptographic functions. When the term “TOE” is used in this section, it refers to each appliance.

The algorithm implementations have been tested in accordance to validation suites set by the Cryptographic Algorithm Validation Program (CAVP) and tested on specific processors. Refer to section 7.4 of this document for the listings of CAVP certificate for each TOE component for each SFR.

The TOE supports data encryption and decryption with AES using CBC and GCM modes with key sizes of 128 and 256. The algorithms supported for cryptographic hashing services are SHA-1, SHA-256, SHA-384 and SHA-512 with message digest sizes of 160, 256, 384 and 512 bits respectively. The algorithms supported for keyed-hash message authentication are HMAC-SHA-1 (block size – 512 bits), HMAC-SHA-256 (block size – 512 bits), HMAC-SHA-384 (block size – 1024 bits) and HMAC-SHA-512 (block size – 1024 bits) with key sizes 160, 256, 384 and 512 bits and message digest sizes of 160, 256, 384 and 512 bits respectively.

The TOE supports RSA, FFC, and ECDSA in the evaluated configuration. RSA and ECDSA digital signature are used in TLS connections and SSH connections (RSA only). The TOE can be configured to use RSA and ECDSA to authenticate IPsec connections.</td></tr>
</table>

TOE SFRs	How the SFR is Satisfied
	Key generation for asymmetric keys on all models of the TOE implements ECDSA with NIST curve sizes P-256, P-384, and P-521 according to FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.4, RSA with key sizes 2048 and 3072 bits according to FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3 and FFC schemes using ‘safe-prime’ groups that meet RFC 3526 and RFC 7919. Asymmetric cryptographic keys used for IKE peer authentication are generated according to FIPS PUB 186-5, Appendix B.3 for RSA schemes and Appendix B.4 for ECDSA schemes. The TOE does not implement any functionality defined as ‘shall not/should not’ and does not omit any functionality related to ‘shall/should’ statements in Appendix B.3 and B.4. There are no TOE-specific extensions, processing or alternative implementations that are not included in the Appendices. Key establishment for asymmetric keys on the TOE implements RSA-based (RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447), ECDSA-based and DH-based key establishment schemes as specified in NIST SP 800-56A “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”. In addition, the TOE also supports DH group 14 key establishment scheme that meets standard RFC 3526, section 3 for interoperability. The TOE's software implementation uses the prime number and generator value specified in RFC 3526 Section 3 when generating parameters for the DH Group 14 key exchange.

Scheme	SFR	Services
RSA	FCS_TLSS_EXT.1, FCS_IPSEC_EXT.1(1), FCS_IPSEC_EXT.1(2)[VPN] FCS_SSHS_EXT.1(1), FCS_SSHS_EXT.1(2), FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.2	HTTPS Remote Administration, SSH Remote Administration, syslog over IPsec, NTP over IPsec, Distributed TOE Communication, Syslog over TLS and IPsec VPN communications
ECC (P-256, P-384, P-521)	FCS_TLSC_EXT.1, FCS_TLSC_EXT.2, FCS_TLSS_EXT.2 FCS_IPSEC_EXT.1(2)[VPN]	Syslog over TLS, Syslog over IPsec, IPsec VPN communications.
ECC (P-256, P-384, P-521)	FCS_TLSS_EXT.1	HTTPS Remote Administration
FFC	FCS_TLSC_EXT.1	Distributed TOE Communication
Diffie-Hellman (Group 14)	FCS_SSHS_EXT.1(1) FCS_SSHS_EXT.1(2) FCS_IPSEC_EXT.1(1) FCS_IPSEC_EXT.1(2)[VPN]	SSH Remote Administration, IKE communication.

The TOE uses a platform-based random bit generator that complies with ISO/IEC 18031:2011 using HMAC_DRBG w/SHA-256 Deterministic Random Bit Generation (DRBG) operating in FIPS mode. In addition, the DRBG is seeded by an entropy source that is at least 256-bit value derived from various highly sensitive and proprietary noise sources described in the proprietary Entropy Design document.

Additionally, the TOE is designed to zeroize secret and private keys when they are no longer required by the TOE. The table in section 7.3 identifies the applicable secret and private keys and summarizes how they are deleted. The secret keys used for symmetric encryption,

TOE SFRs	How the SFR is Satisfied
	private keys, and CSPs used to generate keys, are zeroized immediately after use (for IPsec VPN functions, within FTD only), or on system shutdown (for all other functions). For plaintext keys unrelated to IPsec VPN: the TOE destroys the reference to the keys stored in volatile memory directly followed by a request for garbage collection; the TOE destroys the abstraction that represents the key for keys stored in non-volatile storage the TSF. **FXOS** The TOE utilizes a cryptographic module certificate (i.e., Cisco FIPS Object Module or FOM) to provide supporting cryptographic functions. The algorithm implementations have been tested in accordance to validation suites set by the Cryptographic Algorithm Validation Program (CAVP). Refer to section 7.4 of this document for the listings of CAVP certificate for each TOE component for each SFR. The TOE supports RSA, FFC, and ECDSA in the evaluated configuration. RSA and ECDSA digital signature are used in TLS connections and SSH connections (RSA only). The TOE can be configured to use RSA to authenticate IPsec connections. Key establishment for asymmetric keys on the TOE implements RSA-based (RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447), ECDSA-based and DH-based key establishment schemes as specified in NIST SP 800-56A “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”. In addition, the TOE also supports DH group 14 key establishment scheme that meets standard RFC 3526, section 3 for interoperability. The TOE's software implementation uses the prime number and generator value specified in RFC 3526 Section 3 when generating parameters for the DH Group 14 key exchange. The TOE supports data encryption and decryption with AES using CBC and GCM modes with key sizes of 128, 192 (FXOS only) and 256 (CBC mode only). The algorithms supported for cryptographic hashing services are SHA-1, SHA-256, SHA-384 and SHA-512 with message digest sizes of 160, 256, 384 and 512 bits respectively. The algorithms supported for keyed-hash message authentication are HMAC-SHA-1 (block size – 512 bits), HMAC-SHA-256 (block size – 512 bits), HMAC-SHA-384 (block size – 1024 bits) and HMAC-SHA-512 (block size – 1024 bits) with key sizes 160, 256, 384 and 512 bits and message digest sizes of 160, 256, 384 and 512 bits respectively. The TOE uses a platform-based random bit generator that complies with ISO/IEC 18031:2011 using HMAC_DRBG w/SHA-256 Deterministic Random Bit Generation (DRBG) operating in FIPS mode. In addition, the DRBG is seeded by an entropy source that is at least 256-bit value derived from various highly sensitive and proprietary noise sources described in the proprietary Entropy Design document. For plaintext keys in FXOS the TOE destroys the reference to the keys stored in volatile memory directly followed by a request for garbage collection, and the TOE destroys the abstraction that represents the key for keys stored in non-volatile storage the TSF.
FCS_HTTPS_EXT.1 FCS_TLSC_EXT.1 FCS_TLSC_EXT.2 FCS_TLSS_EXT.1 FCS_TLSS_EXT.2	The TOE implements HTTP over TLS (or HTTPS) to support remote administration on FMC and FXOS, TLS clients to support secure syslog connections, and TLS server and clients to support FPT_ITT.1. A remote administrator can connect over HTTPS to the TOE with their web browser. FTD supports two different TLS clients that send syslog messages to the external syslog server- FTD TLS client and FTD OS TLS Client. When CC mode is enabled, the TOE is restricted to only support TLSv1.2 for HTTPS sessions, client/server communications between TOE components and for syslog communications with AES 128- or 256-bit symmetric ciphers in CBC and GCM modes, in conjunction with

TOE SFRs	How the SFR is Satisfied
	SHA, RSA, and ECDSA. The TOE implements HTTPS according to RFC 2818 by using a TLSv1.2 session to secure the HTTP connection. The TLS cipher suites that are implemented by the TOE are listed in sections 5.3.3.15 and 5.3.3.17. • Relevant to FTP_ITC and FCS_TLSC_EXT.1, **FTD TLS client,** that is configured by the FMC and is the main audit system for audits generated by FTD. It sends audit events such as IPsec and login messages to the external syslog server and Mutual authentication is not supported. Listed in Section 5.3.3.15. • Relevant to FTP_ITC and FCS_TLSC_EXT.2, **FTD OS TLS client**, that is configured through the FTD's command line and sends audit events to an external syslog server such as SSH login, console login, etc. and Mutual authentication is not supported. Listed in Section 5.3.3.16. • Relevant to FTP_ITC and FCS_TLSC_EXT.2, for syslog over TLS from FMC/FMCv (client only) as listed in Section 5.3.3.16 of this document. Mutual authentication is supported. • Relevant to FPT_ITT and FCS_TLSS_EXT.2 (FMC as server and FTD as client) are as listed in sections 5.3.3.18 of this document. • Relevant to FTP_TRP.1/Admin and FCS_TLSS_EXT.1 (server only) are as listed in section 5.3.3.17 of this document (FMC and FXOS) While the cryptographic modules of the TOE support additional cipher suites (for example, RSA_3DES_EDE_CBC_SHA, RSA_DES_CBC_SHA, RSA_RC4_128_MD5, RSA_RC4_128_SHA, etc.), they are all disabled while operating in CC mode. If the TLS client does not support TLSv1.2, the TLS connection will fail and the administrators will not establish a HTTPS web-based session with the TOE. The Key establishment parameters for each of the TLS connections in the TOE are as follows – 1. FMC/FMCv (HTTPS/TLS)- 2048-bit RSA, ECDHE secp256r1, secp384r1 and secp521r1 2. FMC/FMCv and FTD (ITT) – 2048-bit RSA, ECDHE secp256r1, secp384r1 and secp521r1 3. FXOS (HTTPS/TLS) - 2048-bit RSA, ECDHE secp256r1, secp384r1 and secp521r1 When in CC mode and the TOE acts as a TLS client (e.g., connection to the syslog server), the TOE will verify the server Subject Alternative Name (SAN) against the reference identity (wildcard is supported as required in section 6 of RFC 6125 and per RFC 5280 Appendix A. RFC 5280 is supported for the TLS connection between the distributed TOE components (FMC and FTD) and the attribute type "id-at-title" is used by the TOE client to match the presented identifier with the configured identifier). If verification fails, the TLS connection will not be established. The following NIST curves are presented with the Client Hello by default – secp256r1, secp384r1 and secp521r1. Mutual authentication must be configured with the client-side X.509v3 certificate with RSA 2048-bits (or higher) and SHA-256 (or higher). The key agreement parameters of the server key exchange message are specified in the RFC 5246 (section 7.4.3) for TLSv1.2 and the TOE conforms to this RFC. The FMC and FTD must successfully complete a registration process to communicate, which requires administrative actions on the FMC and corresponding administrative actions on the FTD. The administrative actions on FMC and FTD require the administrator to input a

TOE SFRs	How the SFR is Satisfied
	"registration key" that the two devices will use to authenticate their initial TLS communications. During the registration process, the FMC and FTD confirm they have a matching registration key and use their initial self-signed TLS certificates to uniquely identify themselves to each other (each device certificate signed by FMC, including its own, contains a unique identifier stored as an 'id-at-title' attribute, which FMC and FTD each as the unique reference identifier for each other). If the authentication succeeds, the local CA within the FMC will sign and issue a new TLS certificate for the FTD and send (over the existing TLS session) the FTD's new identity certificate and associated keys, and the FMC's root CA cert, and the FMC's root CA certificate and the device certificates which it signed will be used to authenticate all subsequent TLS sessions between the two devices. If device registration fails due to mismatched registration keys, or incorrect IP address or hostname, the information on the FMC and/or FTD needs to be corrected and the registration from FMC reinitiated. TLS session resumption is supported for the following TLS connections of the TOE – the WebUI of the FMC/FMCv and the WebUI of FXOS. The session tickets used for TLS session resumption are encrypted using symmetric algorithms consistent with FCS_COP.1/DataEncryption claims in this ST – AES used in CBC and GCM modes and key sizes of 128 and 256 bits. The session tickets adhere to the structural format provided in section 4 of RFC 5077. For mutual authentication, no fallback authentication for certificates is supported. The TOE will reject the connection when the certificate is deemed invalid.
FCS_IPSEC_EXT.1(1), FCS_IPSEC_EXT.1(2)[VPN]	FTD The IPsec implementation provides VPN peer-to-peer, VPN site-to-site, and VPN client to TOE (i.e., remote access) capabilities. The VPN site-to-site tunnel allows for example the TOE acting as a VPN gateway and another TOE to establish an IPsec tunnel to secure the passing of user data [FTD Only]. Another configuration is the peer-to-peer configuration where the TOE can be set up with an IPsec tunnel with a VPN peer to secure the session between the TOE and the VPN peer [FTD and FXOS]. The VPN client to TOE configuration is where a remote VPN client connects into the TOE in order to gain access to an authorized private network [FTD Only]. Authenticating with the TOE would give the VPN client a secure IPsec tunnel to connect over the internet into their private network. The TOE implements IPsec to provide X509v3 certificate (FTD and FXOS) authentication and encryption services to prevent unauthorized viewing or modification of data as it travels over the external network. The TOE implementation of the IPsec standard (in accordance with the RFCs noted in the SFR) uses the Encapsulating Security Payload (ESP) protocol to provide authentication, encryption and anti-replay services. In addition, the TOE supports both transport and tunnel modes. Transport mode is only supported for peer-to-peer IPsec connection while tunnel mode is supported for all VPN connections including remote access. IPsec Internet Key Exchange, also called IKE, is the negotiation protocol that lets two peers agree on how to build an IPsec Security Association (SA). In the evaluated configuration, only IKEv2 is supported. The IKEv2 protocols implement Peer Authentication using the RSA (FTD and FXOS), ECDSA (FTD Only) algorithm with X.509v3 certificates. IKEv2 separates negotiation into two phases: SA and Child SA. IKE SA creates the first tunnel, which protects later IKE negotiation messages. The key negotiated in IKE SA enables IKE peers to communicate securely in IKE Child SA. During

TOE SFRs	How the SFR is Satisfied
	Child SA IKE establishes the IPsec SA. IKE maintains a trusted channel, referred to as a Security Association (SA), between IPsec peers that is also used to manage IPsec connections, including: • The negotiation of mutually acceptable IPsec options between peers (including signature-based peer authentication parameters), • The establishment of additional Security Associations to protect packet flows using Encapsulating Security Payload (ESP), and • The agreement of secure bulk data encryption AES keys for use with ESP. After the two peers agree upon a policy, the security parameters of the policy are identified by an SA established at each peer, and these IKE SAs apply to all subsequent IKE traffic during the negotiation FXOS implements IPsec using the ESP protocol as defined by RFC 4303, using the cryptographic algorithms AES-CBC-128, AES-CBC-192, AES-CBC-256, and AES-GCM-128 (both specified by RFCs 3602 and 4106) along with SHA-based HMAC algorithms, and using IKEv2, as specified for FCS_IPSEC_EXT.1.5(2)[VPN], to establish security associations. NAT traversal is supported in IKEv2 by default. The IKE SA exchanges use only main mode and the IKE SA lifetimes are able to be limited to 24 hours for Phase 1 (SAs) and 8 hours for Phase 2 (Child SAs). Administrators can configure the mode for each IPsec tunnel, as in the following examples: Devices > VPN > Site To Site or Devices > VPN > Remote Access IKE Options (click on **IKE** tab) IKEv2 Mode **Tunnel mode** — (default) Encapsulation mode is set to tunnel mode. Tunnel mode applies ESP encryption and authentication to the entire original IP packet (IP header and data), hiding the ultimate source and destination addresses and becoming the payload in a new IP packet. **Transport preferred** — Encapsulation mode is set to transport mode with an option to fallback to tunnel mode if the peer does not support it. In Transport mode only the IP payload is encrypted, and the original IP headers are left intact. **Transport required** — Encapsulation mode is set to transport mode only, falling back to tunnel mode is not allowed. If the endpoints cannot successfully negotiate transport mode, due to one endpoint not supporting it, the VPN connection is not made. **Lifetime (seconds)** – The number of seconds a security association exists before expiring is in the range between 120 and 2,147,483,647 seconds The default is 28,800 seconds. **Lifetime (kbytes)** – The volume of traffic (in kilobytes) that can pass between IPsec peers using a given security association before it expires. The range is 10-2,147,483,647 (10KB to 2TB) and the default is 4,608,000 kilobytes. No specification allows infinite data. In the evaluated configuration, use of “confidentiality only” (i.e. using ESP without authentication) for IPsec connections is prohibited. The TOE allows the administrator to define the IPsec proposal for any IPsec connection to use specific encryption methods and authentication methods as in the following objects:

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td></td><td>Objects > Object Management > VPN > IKEv2 IPsec Proposal
Choose Add IKEv2 IPsec Proposal
Enter a **Name**
Enter a Description
Choose **ESP Hash** method from {sha-1 | sha-256 | sha-384 | sha-512 | null}
Choose **ESP Encryption** method from {aes | aes-256 | aes-gcm | aes-gcm-256 ~~| aes-gmac | aes-gmac-192 | aes-gmac-256~~}

Note: When AES-GCM is used for encryption, the ESP integrity selection will be “null” because GCM mode provides integrity. AES-GMAC is not allowed in the evaluated configuration.
The IKEv2 protocols supported by the TOE implement the following DH groups: 14 (2048-bit MODP), 19 (256-bit Random ECP) and 20 (384-bit Random EC), and use the RSA and ECDSA algorithms for Peer Authentication. The following examples are used to specify the DH Group used for SAs:
Objects > Object Management > VPN > IKEv2 Policy
Choose Add IKEv2 Policy
Enter a **Name**
Enter a Description
Enter a Priority
Enter the **Lifetime** of the SA in seconds. You can specify a value from 120 to 2,147,483,647 seconds. The default is 86400.
Choose **Integrity Algorithms** from **[**~~md5 |~~ sha | sha256 | sha384 | sha512**]**
Choose **Encryption Algorithm** from [~~null | des | 3des |~~ aes | ~~aes-192~~ | aes-256 | aes-gcm | ~~aes-gcm-192~~ | aes-gcm-256]
Choose **PRF Algorithm** from {sha | sha256 | sha384 | sha512}
Add a **DH Group** from {14 | 19 | 20 }

The secret ‘x’ generated is 64 bytes long (or 512 bits), is the same across all the DH groups, and is generated with the DRBG specified in FCS_RBG_EXT.1. This is almost double the size of the highest comparable strength value which is 384 bits. The TOE generates nonces used in IKEv2 exchanges, of at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash.
The TOE has a configuration option to deny tunnel if the phase 2 SA is weaker than the phase 1. The crypto strength check is enabled via the **Enable Security Association (SA) Strength Enforcement** checkbox.
The TOE can be configured to authenticate IPsec connections using RSA and ECDSA signatures. When using RSA and ECDSA signatures for authentication, the TOE and its peer must be configured to obtain certificates from the same certification authority (CA).
Devices > VPN > Site To Site or Devices > VPN > Remote Access
IKE Options (click on **IKE** tab)
Policy - Choose a predefined IKEv2 policy object or create a new one to use.</td></tr>
</table>

TOE SFRs	How the SFR is Satisfied

	Authentication Type
	
	• **Certificate** — When you use Certificates as the authentication method for VPN connections, peers obtain digital certificates from a CA server in your PKI infrastructure, and trade them to authenticate each other.
	
	To configure an IKEv2 connection to use a RSA or ECDSA signature, select the authenticate type **Certificate**.
	
	To define rules for matching the DN or FQDN of the IPsec peer certificate:
	
	First, create a certificate map via FMC (Objects > Object Management > VPN > Certificate Map), and add a rule to the certificate map to match the "Alternative Subject" field of the certificate to a value (FQDN/DN)
	
	Next, associate the certificate map with the tunnel, depending on tunnel type:
	
	• Peer-to-peer VPN (Devices > VPN > Site To Site > Add VPN > Firepower Threat Defense Device > add a node > Certificate Map)
	• Remote Access VPN (Devices > VPN > Remote Access > Advanced > Certificate Maps > check "Use the configured rules to match a certificate to a Connection Profile > Add Mapping > Certificate Map Name)
	
	A crypto map (the Security Policy Definition) set can contain multiple entries, each with a different access list. The crypto map entries are searched in a top-down sequence - the TOE attempts to match the packet to the crypto access control list (ACL) specified in that entry. The crypto ACL can specify a single address or a range of address and the crypto map can be applied to an inbound interface or an outbound interface. When a packet matches a permit entry in a particular access list, the method of security in the corresponding crypto map of that interface is applied. If the crypto map entry is tagged as ipsecisakmp, IPsec is triggered. The traffic matching the permit crypto ACLs would then flow through the IPSec tunnel and be classified as PROTECTED. Traffic that does not match a permit crypto ACL or match a deny crypto ACL in the crypto map, but is permitted by other ACLs on the interface is allowed to BYPASS the tunnel. Traffic that does not match a permit crypto ACL or match a deny crypto ACL in the crypto map, and is also blocked by other non-crypto ACLs on the interface would be DISCARDED.
	
	FXOS
	
	When CC mode is enabled, FXOS supports the following:
	
	• **IKE version**: version 2
	• **IPsec Mode**: tunnel, transport
	  o set mode {transport, tunnel}
	• **IKEv2 Mode**: main mode
	• **IKEv2 Ciphers**:
	  o **Encryption algorithms**: AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128
	  o **Integrity algorithms**: SHA-1, SHA-1_160, SHA-256, SHA384, and SHA-512

TOE SFRs	How the SFR is Satisfied
	◦ **DH Groups**: 14, 15, 16, 19, 20 and 21 • **ESP Ciphers**: ◦ **Encryption algorithms:** AES-CBC-128, AES-CBC-192, AES-CBC-256, AES-GCM-128 ◦ **Integrity algorithms:** SHA-1, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512 • **Authentication**: X.509v3 certificates ◦ create authority *trustpoint_name* • **Traffic Selector**: remote host or subnet ◦ set local-addr *ip_address* ◦ set remote-addr *ip_address* ◦ set remote-subnet *ip/mask* ◦ set remote-ike-ident *remote_identity_name* • **IKEv2 SA Life Time:** Configurable within 60-1440 minutes, including 24 hours. ◦ set ike-rekey-time *minutes* • **IKEv2 Child SA Life Time:** Configurable within 30-480 minutes, including 8 hours. ◦ set esp-rekey-time *minutes* • **Reference Identifier** ◦ set remote-ike-ident *remote_identity_name* The secret 'x' generated is 64 bytes long (or 512 bits), and is generated with the DRBG specified in FCS_RBG_EXT.1. This is almost double the size of the highest comparable strength value which is 384 bits. The TOE generates nonces used in IKEv2 exchanges, of at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash. The TOE has a configuration option to deny tunnel if the phase 2 SA is weaker than the phase 1. In FXOS, the SPDs are pretty simple because FXOS is not operating as a VPN gateway, and the SPDs are just based on IP addresses, so the type of traffic being tunneled (e.g. syslog) is irrelevant to the tunneling decisions. • The local-addr is the local management IP. • The remote-addr is the IP of the IPsec peer (in tunnel mode or transport mode). • A remote-subnet is applicable only in tunnel mode, and defines the subnet that would be reachable beyond the remote-addr. • Outbound traffic will be **encrypted** when the source address is local-addr, ***and***: ◦ the destination address is the remote-addr (in tunnel or transport mode); ***or*** ◦ the destination address is on the remote-subnet (in tunnel mode). • Outbound traffic will **bypass** the tunnel if: ◦ the destination address is ***not*** the remote-addr; ***and***

TOE SFRs	How the SFR is Satisfied
	◦ the destination address is **not** on the remote-subnet. • Inbound traffic will be **dropped** if: ◦ the source address (prior to decryption) is on the remote-subnet (in tunnel mode); **or** ◦ the source address is the remote-address, **and** the packets are **not** IKE or ESP. To configure an IPsec connection, rules need to be defined for matching the DN of the IPsec peer certificate.
FCS_NTP_EXT.1(1), FCS_NTP_EXT.1(2)	Administrators can update the TOE's clock manually via FXOS or FMC and can also configure the TOE (FXOS and FMC) to use NTP to synchronize the TOE's clock with an external time source. The FTD automatically synchronizes its clock with the FXOS clock. NTPv3 is supported by the FXOS and the NTP timestamp is not updated from broadcast or multicast addresses. IPsec is used to secure the connection between the FXOS and the NTP time source. Administrators can update the TOE's clock manually via FMC or can configure the TOE to use NTP to synchronize the TOE's clock with an external time source. NTPv4 is supported by the FMC and the NTP timestamp is not updated from broadcast or multicast addresses. Authentication using SHA-1 as the message digest algorithm is used to secure the NTP updates.
FCS_SSHS_EXT.1(1)	**FXOS** FXOS implement SSHv2 servers as specified in RFCs 4252 and 4253 (telnet is disabled in the evaluated configuration). For 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. Rekey occurs after any of the thresholds are reached. SSH connections will be dropped if the TOE receives a packet larger than 262,126 bytes. The FXOS's implementation of SSHv2 supports: • Public key algorithms rsa-sha2-256, rsa-sha2-512 and ecdsa-sha2-nistp384 for signing and verification as part of the SSH authentication. • Password-based authentication for administrative users. • Encryption algorithms, AES-CBC-128, AES-CBC-256 and AEAD-AES-256-GCM to ensure confidentiality of the session. • Hashing algorithm hmac-sha1 to ensure the integrity of the session for FXOS. FXOS additionally supports hmac-sha2-256, hmac-sha2-512 and AEAD-AES-256-GCM. • Requiring use of DH group 14 and ecdsa-sha2-nistp384 as key exchange methods. • The TOE verifying the SSH client's presented public key matches one stored within the TOE's SSH server's authorized keys file. • SSH user public key-based authentication using ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384 and ecdsa-sha2-nistp521

TOE SFRs	How the SFR is Satisfied
	FXOS allows authorized administrator to configure the maximum data and maximum time in compliance with the limits as specified in FCS_SSHS_EXT.1.8(1), such that the rekey will occur whichever threshold (data or time) is hit first. MIO-A /system/services # **set ssh-server rekey-limit volume** *[KB]* **time** *[Minutes]*
FCS_SSHS_EXT.1(2)	**FMC and FTD** The TOE supports SSHv2 with the following encryption algorithms - aes128-cbc, aes256-cbc, AEAD_AES_128_GCM, AEAD_AES_256_GCM, in conjunction with HMAC-SHA1, HMAC-SHA-256, HMAC-SHA-512, AEAD_AES_128_GCM and AEAD_AES_256_GCM for integrity and authenticity, and ecdh-sha2-nistp384 and with diffie-hellman-group14-sha1 (supported only by FTD), ecdh-sha2-nistp256 (supported only by FTD), ecdh-sha2-nistp384, and ecdh-sha2-nistp521 (supported only by FTD) for the key exchange methods. While DES and 3DES, HMAC-MD5 and HMAC-MD5-96, and diffie-hellman-group-1 and other diffie-hellman-exchange groups are all implemented, they are disabled while the TOE is operating in CC Mode. In addition, SSHv1 is also disabled by default for security reasons. If the SSH client does not support the Approved algorithms or SSH version, the SSH connection will fail and the administrators will not establish an SSHv2 CLI session with the TOE. The TOE supports SSH host key authentication using rsa-sha2-256, rsa-sha2-512, and ecdsa-sha2-nistp384 algorithms and SSH user public key-based authentication using ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384 and ecdsa-sha2-nistp521. The TOE ensures and verifies that the SSH client's presented public key matches one that is stored within the TOE's SSH server's authorized keys file. The TOE also supports password-based user authentication for SSH. The TOE uses OpenSSH implementation version 9.1p1 to support the SSHv2 connections. The authentication timeout period is 90 seconds allowing clients to retry only 3 times. In addition, both public-key (RSA) and password-based authentication can be configured with password-based being the default method used. Whenever the timeout period or authentication retry limit is reached, the TOE closes the applicable TCP connection and releases the SSH session resources. As SSH packets are being received, the TOE uses a buffer to build all packet information. Once complete, the packet is checked to ensure it can be appropriately decrypted. However, if it is not complete when the buffer becomes full (256 Kbytes) the packet will be dropped. Note that the TOE manages a tracking mechanism for each SSH session so that it can initiate a new key exchange when either approximately 1 hour of time or 1GB of data is reached. An audit event is generated when a successful SSH rekey occurs when either of the thresholds mentioned occurs. SSH connections will be dropped if the TOE receives a packet larger than 262126 bytes.
FIA_AFL.1	**FMC** FMC provides the administrator the ability to specify the maximum number (can be set differently per account on FMC) of unsuccessful authentication attempts via SSH or WebUI (default is five attempts, configurable from 1-999) before the offending account is locked. The configured limit is the maximum number of allowed consecutive failures, thus the defined number of unsuccessful consecutive authentication attempts that results in locking of accounts is one more than the maximum number of allowed consecutive failures. Only an

TOE SFRs	How the SFR is Satisfied
	authorized administrator (with the 'administrator' role) can unlock a locked account. By default, the predefined 'admin' account is exempt from becoming locked, but that default is overridden when CC mode is enabled. If all admin accounts become locked for any reason, FMC can be accessed locally using password recovery procedures. **FXOS** FXOS will allow a maximum number (same value applies to all FXOS accounts) of consecutive failed login attempts via SSH or WebUI before the offending account becomes locked ('lock-status' set to 'locked'). When an account is locked, it can be unlocked by another administrator who has the 'admin' role (not just 'read-only'). An account can also be configured to be unlocked after an administrator-defined time period without the intervention of another administrator with 'admin' role. If all admin accounts become locked for any reason, FXOS can be accessed locally using password recovery procedures. • All types of user accounts (including account type 'admin') are locked out of the system after exceeding the maximum number of login attempts. • The default maximum number of unsuccessful login attempts is '3' (configurable from 1-10). **FTD** The FTD CLI provides the administrator the ability to specify the maximum number of unsuccessful authentication attempts (configurable from 1-9999) before the offending account is locked. Only an authorized administrator (with the 'administrator' role) can unlock a locked account. An account can also be configured to be unlocked after an administrator-defined time period without the intervention of another administrator with 'admin' role. If all admin accounts become locked for any reason, FTD can be accessed locally using password recovery procedures. The FTD CLI also provides the administrator the ability to specify the time duration for which a locked account remains locked (configurable from 1 to 9999 minutes). Once an account is locked, it automatically unlocks after the time-period that the administrator has configured, has elapsed.
FIA_PMG_EXT.1	The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper and lower-case letters, numbers, and special characters as listed in the SFR (FXOS passwords do not support "=" or "$" characters). Minimum password length is settable by the Authorized Administrator, and support passwords of 8 to 32 characters (FTD and FMC) and 8 to 127 characters (FXOS) when "enforce-strong-password" option is enabled in security scope. Password composition rules specifying the types and number of required characters that comprise the password are settable by the Authorized Administrator. Passwords can be configured with a maximum lifetime, configurable by the Authorized Administrator. New passwords can be required to contain a minimum of 4-character changes from the previous password.
FIA_UIA_EXT.1	The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed. All the TOE components support authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2. Administrative access to the TOE is facilitated through the TOE's CLI (SSH (password-based and public key-based) or local console in FTD, FMC and FXOS), or web GUI in FMC and FXOS. The TOE mediates all administrative actions through the CLI and GUI. The TOE presents a warning banner in accordance with FTA_TAB.1 requirement prior to initiating the identification

TOE SFRs	How the SFR is Satisfied
	authentication mechanism for those attempting to access the TOE. Once a potential administrative user attempts to access an administrative interface either locally or remotely, the TOE prompts the user for a username and password. For remote SSH connections, an administrative user can alternately use a username and an SSH public/private keypair for authentication rather than a username and password combination. Only after the administrative user presents the correct authentication credentials will access to the TOE administrative functionality be granted. No access is allowed to the administrative functionality of the TOE until an administrator is successfully identified and authenticated. The TOE provides an automatic lockout when a user attempts to authenticate and enters invalid credentials. After a defined number of authentication attempts fail exceeding the configured allowable attempts, the user is locked out until an authorized administrator can unlock the user account. FTD and FXOS also support the configuration for an account to be unlocked after an administrator-defined time period without the intervention of another administrator with 'admin' role.
FIA_UAU_EXT.2	The TOE provides local password-based authentication mechanisms to FMC, FTD and FXOS. The process for authentication is the same for administrative access whether administration is occurring via a directly connected console cable or remotely via SSHv2 (password-based or public key-based) or TLS. At initial login in the administrative user is prompted to provide a username. After the user provides the username, the user is prompted to provide the administrative password associated with the user account. The TOE then either grants administrative access (if the combination of username and password is correct) or indicates that the login was unsuccessful. The TOE does not provide indication of whether the username or password was the reason for an authentication failure. FXOS also provides support for RADIUS and TACAC+ authentication mechanisms. For SSH connections, an administrative user can also present an SSH private key along with a username, rather than a username/password combination. The SSH private key must correspond to an SSH public key that is loaded on the TOE by following the instructions found in the AGD. After the user provides the username and SSH private key, the TOE then either grants administrative access (if the private key corresponds to a public key known to the TOE and the username is correct) or will prompt for a password, indicating the SSH public/private keypair login was unsuccessful. Similarly in this case, the TOE does not provide an indication whether the username or SSH private key was the reason for authentication failure.
FIA_UAU.7	When logging in, the TOE will not echo passwords such that passwords are not inadvertently displayed to the user and any other users that might be able to view the login display. The TOE replaced the entered password character with a "*" character or not show any character at all. This depends on where the user is logging in from, for example, using web GUI versus the SSH client. If the authentication fails, the TOE is designed to not indicate either the username and/or password were incorrect. The error message would just state access denied or unable to authorize access. No other information about the failed login in can be ascertained from the error message. Note also that should a user have their session terminated (e.g., due to inactivity), they are required to successfully re-authenticate, by re-entering their identity and authentication

TOE SFRs	How the SFR is Satisfied
	data, in order to gain access to their session. The authentication data is not cached by the TOE for any reason.
FIA_X509_EXT.1/ITT FIA_X509_EXT.1/Rev FIA_X509_EXT.2(1) FIA_X509_EXT.2(2) FIA_X509_EXT.3	The TOE support X.509v3 certificates as defined by RFC 5280. Public key infrastructure (PKI) credentials, such as private keys and certificates are stored securely. The identification and authentication, and authorization security functions protect an unauthorized user from gaining access to the storage. The validity check for the certificates takes place at session establishment and/or at time of import depending on the certificate type. For example, server certificate is checked at session establishment while CA certificate is checked at both. The TOE conforms to standard RFC 5280 for certificate and path validation (i.e., peer certificate checked for expiration, peer certificate checked if signed by a trusted CA in the trust chain, peer certificate checked for unauthorized modification, peer certificate checked for revocation). The TOE can generate a RSA key pair that can be embedded in a Certificate Signing Request (CSR) created by the TOE. The CSR can be generated at the UI. The TOE can then send the CSR manually to a Certificate Authority (CA) for the CA to sign and issue a certificate. Once the certificate has been issued, the administrator can import the X.509v3 certificate into the TOE. Integrity of the CSR and certificate during transit are assured through the use of digital signature (signing the hash of the TOE's public key contained in the CSR and certificate). CRL is configurable and can be used for certificate revocation check (for FTP_ITC only, thus relevant only to FIA_X509_EXT.1/Rev, not relevant to FIA_X509_EXT.1/ITT as no revocation checking is used for communications between TOE components). Checking is also done for the 'basicConstraints' extension and the 'cA' flag to determine whether they are present and set to TRUE. If they are not, the CA certificate is not accepted as a trust anchor. FMC and FXOS only support CRL, while FTD supports use of both CRL and OCSP (including verification of the OCSP signing purpose in the certificate that signs the OCSP response). FTD supports CRL for other purposes, i.e., for validation of syslog server certificates for both the TLS connections to TLS servers. The administrators can configure a trust chain by importing the CA certificate(s) that signed and issued the server (syslog) certificate. This will tell the TOE which CA certificate(s) to use during the validation process. If the TOE does not find the trusted root CA, the TLS connections (FTD TLS client and FTD OS TLS client) to the syslog server will fail. When the TOE cannot establish a connection for the validity check using CRL or the OCSP responder for verification, the FTD and FMC will not accept the certificate. When communicating with peers, the TOE uses the default certificate that is configured through the FMC and one that matches the peer's request. For more information, please refer to the CC Supplemental User Guide. The FXOS does not accept the certificate when the validity check using CRL cannot be completed.
FMT_MOF.1/ManualUpdate FMT_MOF.1/Services	The TOE restricts the ability to enable, disable, determine and modify the behavior of all of the security functions of the TOE to authorized administrators. The TOE provides the ability for authorized administrators to enable or disable service and features, initiate TOE update, access TOE data, such as audit data, configuration data, security attributes, information flow rules, and session thresholds. **FMC** Only accounts with 'administrator' privilege can upload patches to FMC and initiate installation of patches to FMC or FTD devices (the FMC WebUI is used to manually initiate

TOE SFRs	How the SFR is Satisfied
	updates to FMC and FTD). Only accounts with 'administrator' privilege can update system configuration settings related to: • local logging and remote logging • clock settings • account management including account lockout settings and unlocking accounts (for FMC accounts only) • login banners • cryptographic functionality including SSH (FMC and FTD), TLS (FMC), and IPsec (FTD) • generation of CSRs, and import or delete X.509v3 certificates • firewall functionality • VPN functionality • IPS functionality **FTD** Only accounts with 'config' privilege can update system configuration settings related to: • account management including account lockout settings and unlocking accounts (for FTD accounts only) **FXOS** Only accounts with 'admin' role can upload software updates to FXOS and initiate updates of FXOS and configure: • local logging and remote logging • clock settings • account management including account lockout settings and unlocking accounts • login banners • cryptographic functionality including SSH, TLS, and IPsec
FMT_MTD.1/CoreData	**FTD and FMC** The TOE provides a web-based GUI (using HTTPS) management interface and CLI (using SSH or serial connection) (FMC provides the Web GUI and CLI, while the FTD provides a CLI) for all TOE administration, including the policy rule sets, user accounts and roles, and audit functions. The ability to manage various security attributes, system parameters and all TSF data is controlled and limited to those users who have been assigned the appropriate administrative role and privileges associated with those roles. Note that all users created are TOE administrators **Predefined User Roles** The TOE supports the following predefined user roles: • **Administrators** can set up the appliance's network configuration, manage user accounts, and configure system policies and system settings. The Administrator Role provides access to analysis and reporting features, rule and policy configuration, system management, and all maintenance features. Users with the Administrator role have ALL access rights. Note: The only TOE user role is "Administrator". This role is granted when a new user account is created and cannot be changed.

TOE SFRs	How the SFR is Satisfied
	In addition to the all-powerful "Administrator" role, these additional roles help with the management of the IPS policies and functionality - • "IPS Administrator" (or Administrator): Have all privileges and access • "IPS Analyst" (or Intrusion Admin): Have all access to intrusion policies, IPS policies and network analysis privileges but cannot deploy policies • Access Admin: Have all access to access control policies but cannot deploy policies • Discovery Admin: Have all access to network discovery, application detection, and correlation features but cannot deploy policies • Security Analyst: Have all access to security event analysis feature The web-based GUI is available on the FMC. The web-based GUI on the FMC is highly recommended for daily management of the FMC and its managed FTD. Local access to the shell which allows access to the underlying operating system is allowed in the CC evaluated configuration for the initial configuration only. For normal daily operations, the web GUI is still the recommended method. **FXOS** User accounts are used to access the FXOS system through the FXOS WebUI and CLI. Up to 48 local user accounts can be configured. Each user account must have a unique username and password. The 'admin' account is a default user account and cannot be modified or deleted. This account is the system administrator or superuser account and has full privileges. The term "authorized administrator" or "Security Administrator" applies to this account and other accounts assigned to the Administrator role.
FMT_MTD.1/CryptoKeys	The TOE only provides the ability for authorized administrators to access TOE data, such as audit data, configuration data, security attributes (such as cryptographic keys and certificates used in VPN), routing tables, and session thresholds.
FMT_SMF.1 FMT_SMF.1/FFW[FW] FMT_SMF.1/VPN[VPN]	The TOE includes the functions necessary to administer the TOE locally and remotely via the administrative interfaces of the FTD (SSH CLI, local console), FMC (WebUI, SSH CLI, local console) and FXOS (WebUI, SSH CLI and local console). All the management functions that are available to be performed on the TOE local console can also be performed remotely via SSH. No access or service is provided prior to identification and authentication, beyond viewing the login banner. **FMC** FMC administrators can perform the following functions: • Login locally via console CLI, and remotely via SSH CLI or TLS WebUI • Configure the access banner (via WebUI) • Ability to start and stop services via SSH CLI or TLS WebUI • Configure session inactivity time limits (via WebUI) • Update the FMC and FTD TOE components and verify updates using digital signature prior to installing updates (via WebUI) • Configure authentication failure parameters (via WebUI) • Manage cryptographic keys (via WebUI) • Configure cryptographic functionality (via WebUI) • Configure lifetime for IPsec SAs (via WebUI)

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td></td><td>

Import X.509v3 certificates (via WebUI)
Configure interaction between TOE components (via WebUI)
Re-enable an administrator account (via WebUI)
Set the time which is used for time-stamps (via WebUI)
Configure the reference identifier for the peer (via WebUI)
Configure firewall rules (via WebUI)
Define packet filtering rules for VPNs (via WebUI)
Manage the trusted public keys database
Associate packet filtering rules to network interfaces for VPNs (via WebUI)
Configure ordering/sequencing of packet filtering rules (via WebUI)

<u>FTD</u>
FTD administrators can perform the following functions:

Login locally via console CLI, and remotely via SSH CLI
Configure authentication failure parameters
Import X.509v3 certificates (for syslog servers only)
Configure interaction between TOE components
Re-enable an administrator account
Configure the reference identifier for the peer (for syslog servers only)
Manage the trusted public keys database

<u>FXOS</u>
FXOS administrators can perform the following functions:

Login locally via console CLI, and remotely via SSH CLI or TLS WebUI
Configure the access banner (via CLI)
Configure session inactivity time limits (via CLI)
Update the FXOS TOE component and verify updates using digital signature prior to installing updates (via CLI or WebUI)
Configure authentication failure parameters (via CLI)
Manage cryptographic keys (via CLI)
Configure cryptographic functionality (via CLI or WebUI)
Configure lifetime for IPsec SAs (via CLI)
Import X.509v3 certificates (via CLI)
Re-enable an administrator account (via CLI or WebUI)
Set the time which is used for time-stamps (via CLI or WebUI)
Configure NTP (WebUI)
Configure the reference identifier for the peer (via CLI)
Manage the trusted public keys database (via CLI)

</td></tr>
<tr><td>FMT_SMR.2</td><td>
<u>FTD and FMC</u>
The TOE includes one evaluated role which corresponds to the required ‘Security Administrator’ described in Section 5.3.5.6.
<u>FXOS</u>
The system contains the following user role:
Administrator
</td></tr>
</table>

TOE SFRs	How the SFR is Satisfied
	Complete read-and-write access to the entire system. The default admin account is assigned this role by default and it cannot be changed.
FPT_SKP_EXT.1	**FTD and FMC** The TOE is designed to not to disclose or store plaintext passwords (e.g., passwords are never recorded in the audit records or display during authentication process). The passwords are stored hashed using Approved SHA-512 with a 32-bit salt value. Only 'root' user account with access to the root shell can view the hashed passwords and this is prohibited in the evaluated configuration. The same is true for cryptographic keys such as encryption symmetric keys and private keys. The public keys can be viewed but cannot be modified without detection. Note that access to public keys is restricted to administrators. **FXOS** All keys are stored on volatile memory without encryption. Only admin users can load a debugging plugin (which is NOT given to customers) to have a file system based access to key files.
FPT_APW_EXT.1	**FTD and FMC** The TOE is designed to not to disclose or store plaintext passwords (e.g., passwords are never recorded in the audit records or display during authentication process). The passwords are stored hashed using Approved SHA-512 with a 32-bit salt value. Only 'root' user account with access to the root shell can view the hashed passwords and this is prohibited in the evaluated configuration. The same is true for cryptographic keys such as encryption symmetric keys and private keys. The public keys can be viewed but cannot be modified without detection. Note that access to public keys is restricted to administrators. **FXOS** All passwords are stored in hashed form using SHA-512.
FPT_STM_EXT.1	The FMC and FXOS provides a source of date and time information for the TOE, used in audit timestamps, in validating service requests, determining the validity of certificates, and for tracking time-based actions related to session management including timeouts for inactive administrative or remote VPN sessions (FTA_SSL*), and renegotiating SAs for IPsec tunnels (FCS_IPSEC_EXT.1(1)). This function can only be accessed from within the configuration exec mode via the privileged mode of operation or using the appropriate role. The clock function is reliant on the system clock provided by the underlying hardware. The clock's date and time can be adjusted by authorized administrators. FMC and FMCv's clock can be configured manually by the administrators and can also synchronize time with a NTP server. FXOS can either set its time manually or sync with an NTP server. The FTD automatically synchronizes its clock with the FXOS clock.
FPT_TST_EXT.1	The FTD, FMC and FXOS run a suite of self-tests during initial start-up (power-on-self-tests or POST) to verify its correct operation. When CC mode is enabled on the FMC, FTD and FXOS, additional cryptographic tests and software integrity test will be run during start-up. The self-testing includes cryptographic algorithm tests (known-answer tests) that feed pre-defined data to cryptographic modules and confirm the resulting output from the modules match expected values, and firmware integrity tests that verify the digital signature of the code image using RSA-2048 with SHA-512. The cryptographic algorithm testing verifies proper operation of encryption functions, decryption functions, signature padding

TOE SFRs	How the SFR is Satisfied
	functions, signature hashing functions, and random number generation. The firmware integrity testing verifies the FTD, FMC and FXOS images have not been tampered with or corrupted. If any of these self-tests fails, the TOE will cease operation. Noise source health tests are run both periodically and at start-up on the FTD to determine the functional health of the noise source. These tests are specifically designed to catch catastrophic losses in the overall entropy associated with the noise source. Tests are run on the raw noise output, before the application of any conditioners. If a noise source fails the health test either at start-up or after the device is operational, the platform will be shut down. Whenever a failure (e.g., POST or integrity test fails) occurs within the FTD that results in the FTD ceasing operation, the FTD securely disables its interfaces to prevent the unintentional flow of any information to or from the FTD and reloads. So long as the failures persist, the FTD will continue to reload. This functionally prevents any failure from causing an unauthorized information flow. There are no failures that circumvent this protection. The tests are sufficient to ensure the correct operation of the security features because they address integrity of the TOE executing firmware and verify the correctness of the cryptographic operations (including noise source) underlying the security features described by the Security Target.
FPT_TUD_EXT.1	The TOE components (FMC, FTD and FXOS) have specific versions that can be queried by an administrator. When updates are made available by Cisco, an administrator can obtain and manually install those updates. Digital signatures (RSA) , with key sizes of 2048 and 3072 bits, are used to verify software/firmware update files (to ensure they have not been modified from the originals distributed by Cisco) before they are used to update the applicable TOE components. The update process will fail if the digital signature verification process fails. Updates can be downloaded from http://www.cisco.com/go/firepower9300-software or http://www.cisco.com/go/firepower4100-software or https://software.cisco.com with a Cisco.com account. The appropriate software image is then downloaded to the administrator's workstation, then uploaded to FMC, or FXOS (FTD updates are uploaded to FMC then pushed from FMC to FTD). Software update files are verified using digital signatures (RSA) automatically at the time they are uploaded to FMC or FXOS. Update files will fail to be stored on the device if they fail validation. Images stored on FXOS can be re-verified with the command – "verify platform-pack version *version_number*". On FMC, the FMC and FTD updates can uploaded and installed by navigating to System > Updates. Several upload files can remain stored locally on FMC and installed to FMC or FTD at a later time. When updates are initiated they are applied immediately, and the FMC or FTD will reload automatically with the new software version. That same page also shows the currently running version on FMC. To view the currently running version of any FTD, navigate to Devices > Device Management > then select the device > click on the 'Device' tab. On FXOS, FXOS updates can be uploaded by navigating to System > Updates. Several upload files can remain stored locally on FXOS and installed at a later time. When updates are initiated they are applied immediately, and the FXOS will reload automatically with the new software version. To view the currently running version of FXOS, click on the 'Overview' tab.

TOE SFRs	How the SFR is Satisfied
FPT_ITT.1	The communication between the FMC and FTD is protected by TLSv1.2. TLS provides authentication, key exchange, encryption and integrity protection of all data transmitted between the TOE components.
FTA_SSL_EXT.1 FTA_SSL.3	An administrator can configure maximum inactivity times for both local and remote administrative sessions. When a session is inactive (i.e., no session input) for the configured period of time the TOE will terminate the session, requiring the administrator to log in again to establish a new session when needed. The inactivity times are set at a default of 60 minutes, but an Administrator can configure the inactivity time for the FMC and FTD through the FMC WebUI and FXOS through the FXOS CLI.
FTA_SSL.4	An administrator is able to exit out of both local and remote administrative sessions of the FMC, FTD and FXOS, effectively terminating the session so it cannot be re-used and will require authentication to establish a new session.
FTA_TAB.1	The TOE provides administrators with the capability to configure advisory banner or warning message(s) that will be displayed prior to completion of the logon process at the local console or via any remote connection (e.g., SSH or HTTPS). The TOE displays an advisory notice and a consent warning message for each administrative method of access: • FMC/FMCv: Console, SSH, and WebUI • FXOS: Console, SSH, and WebUI • FTD: The FTD CLI (SSH), which provides the login banner.
FTP_ITC.1	The TOE uses IPsec and/or TLS to protect communications between itself and remote entities for the following purposes: • The TOE protects transmission of audit records when sending syslog message to a remote audit server by transmitting the messages: o From FMC/FMCv as a TLS client, using X.509v3 certificates for assured identification of the syslog server and with mutual authentication supported. o From FXOS over IPsec, using X.509v3 certificates for assured identification of the syslog server. o From FTD as a TLS client (FTD TLS Client), that is configured by the FMC and is the main audit system for audits generated by FTD. It sends audit events such as IPsec and login messages to the external syslog server and Mutual authentication is not supported. o From FTD as a TLS client (FTD OS TLS Client), that is configured through the FTD's command line and sends audit events to an external syslog server such as SSH login, console login, etc. and Mutual authentication is not supported. o From FTD to an external syslog server over IPsec. • The TOE protects communication with a NTP server: o From FXOS to a NTP server over IPsec, using RSA for peer authentication that X509v3 certificates.

TOE SFRs	How the SFR is Satisfied
	• The TOE (FTD only) protects peer-to-peer VPN connections between itself and VPN peers (connections can be initiated by the TOE or by the peer) using IPsec, using X.509v3 certificates for assured identification of the peer. • The TOE (FTD Only) protects VPN connections inbound from VPN clients using IPsec, using X.509v3 certificates for assured identification of the VPN client. Note that the remote VPN client is in the operational environment. • Connections to authentication servers (AAA servers) can be protected via IPsec tunnels. Connections with AAA servers can be configured for authentication of TOE administrators. o RADIUS over IPsec (FXOS) o TACACS+ over IPsec (FXOS)
FTP_TRP.1/Admin	The TOE uses SSHv2 or HTTPS to provide the trusted path (with protection from disclosure and modification) for all remote administration sessions. Optionally, the FXOS and FTD support tunneling the SSH and HTTPS connections in IPsec VPN tunnels (remote VPN client). Remote administration of FMC can be performed using SSH or TLS/HTTPS. Remote administration of FXOS can be performed using SSH or TLS/HTTPS. Remote administration through the CLI of FTD is via SSH.
Security Functional Requirements Drawn from mod_ips_v1.0	
FAU_GEN.1/IPS[IPS] FAU_SAR.1[IPS] FAU_SAR.2[IPS] FAU_SAR.3[IPS] FAU_STG.1[IPS]	**FTD and FMC** For each possible intrusion identified by the system, the TOE will generate an event log, also referred to as an intrusion event and event types are not combined. Each event log will include a record of the date, time, type of exploit, and contextual information about the source of the attack and its target. For packet-based events, a copy of the packet or packets that triggered the event is also recorded. Managed Sensors will transmit their events to the FMC where the administrators can view the aggregated data and gain a greater understanding of the attacks against the entire network. The administrators can also deploy the managed Sensors in inline allowing them to configure the Sensors to drop or modify packets that are harmful. The web-based UI is the only way to view the intrusion events (Analysis > Intrusions > Events). The list below describes the intrusion event information that can be viewed, searched, filtered, and sorted by the system. In addition, basic contents such as date, time, and type can also be used to filter and sort. Note only Administrators and Intrusion Admins have access to the intrusion events. **Access Control Policy** The access control policy associated with the intrusion policy where the intrusion, preprocessor, or decoder rule that generated the event is enabled. **Access Control Rule** The access control rule that invoked the intrusion policy that generated the event. Default Action indicates that the intrusion policy where the rule is enabled is not associated with a

TOE SFRs	How the SFR is Satisfied
	specific access control rule but, instead, is configured as the default action of the access control policy. This field is blank if intrusion inspection was associated with neither an access control rule nor the default action, for example, if the packet was examined by the default intrusion policy. **Application Protocol** The application protocol, if available, which represents communications between hosts detected in the traffic that triggered the intrusion event. **Application Risk** The risk associated with detected applications in the traffic that triggered the intrusion event: Very High, High, Medium, Low, and Very Low. Each type of application detected in a connection has an associated risk; this field displays the highest risk of those. **Count** The number of events that match the information that appears in each row. Note that the Count field appears only after you apply a constraint that creates two or more identical rows. This field is not searchable. **Destination Continent** The continent of the receiving host involved in the intrusion event. **Destination Country** The country of the receiving host involved in the intrusion event. **Destination IP** The IP address used by the receiving host involved in the intrusion event. **Destination Port / ICMP Code** The port number for the host receiving the traffic. For ICMP traffic, where there is no port number, this field displays the ICMP code. **Destination User** The User ID for any known user logged in to the destination host. **Device** The managed Sensor where the access control policy was deployed. **Domain** The domain of the Sensor that detected the intrusion. This field is only present if you have ever configured the Firepower Management Center for multitenancy. **Egress Interface** The egress interface of the packet that triggered the event. This interface column is not populated for a passive interface. **Egress Security Zone**

TOE SFRs	How the SFR is Satisfied
	The egress security zone of the packet that triggered the event. This security zone field is not populated in a passive deployment. **Generator** The component that generated the event. **Ingress Interface** The ingress interface of the packet that triggered the event. Only this interface column is populated for a passive interface. **Ingress Security Zone** The ingress security zone of the packet that triggered the event. Only this security zone field is populated in a passive deployment. **Inline Result** Actions **Intrusion Policy** The intrusion policy where the intrusion, preprocessor, or decoder rule that generated the event was enabled. **Message** The explanatory text for the event. For rule-based intrusion events, the event message is pulled from the rule. **Priority** The event priority as determined by the Cisco Talos Security Intelligence and Research Group (Talos). The priority corresponds to either the value of the priority keyword or the value for the classtype keyword. For other intrusion events, the priority is determined by the decoder or preprocessor. Valid values are high, medium, and low. **Protocol (search only)** The name or number of the transport protocol used in the connection. **Signature ID** The signature used to generate the event. **Snort ID (search only)** Specify the Snort ID (SID) of the rule that generated the event or, optionally, specify the combination Generator ID (GID) and SID of the rule, where the GID and SID are separated with a colon (:) in the format GID:SID. **Source Continent** The continent of the sending host involved in the intrusion event. **Source Country** The country of the sending host involved in the intrusion event.

TOE SFRs	How the SFR is Satisfied
	Source IP The IP address used by the sending host involved in the intrusion event. **Source Port / ICMP Type** The port number on the sending host. For ICMP traffic, where there is no port number, this field displays the ICMP type. **Source User** The User ID for any known user logged in to the source host. The intrusion events cannot be modified but they can be deleted by the Administrators or Intrusion Admins who have restricted access. When the intrusion events storage is full, the newest data will overwrite the oldest data. There is a feature called Threshold where the administrators can control the number of events that are generated per rule over time. They can limit notification to the specified number of event instances per time period or provide notification once per time period after a specified number of event instances. The administrator must specify if the event instances will be tracked by source or destination IP address, the count or the number of event instances, and the number of seconds for the time period for which event instances are tracked. Note the IPS function cannot be disabled unless the whole system is shutdown. The TOE also will generate all of the required auditable events identified in Table 22 (for FMT_SMF.1/IPS and IPS_NTA_EXT.1 only). All other events in the table are addressed by intrusion events, not auditable events. Please see the CC Supplemental User Guide for more details. The TOE can be configured to generate intrusion events. In addition, all management functions are audited as well. There are certain header fields that should not be used to trigger intrusion events (in Inline mode or Passive mode). Logging events related to these fields would generate a deluge of intrusion audit records that would prevent IPS analysts from figuring out what security incidents occur in their monitored network. In addition, logging these fields will provide no benefits. Per mod_ips_v1.0, the following fields can be inspected and if in inline mode, dropped or modified (i.e., normalized): − All checksum fields − TCP Reserved field − TCP Urgent Pointer field In inline mode, the TOE can count invalid checksum packets that are dropped. The TOE can also count the packets that gets normalized or dropped because of failed normalization.
FMT_SMF.1/IPS[IPS]	**FMC and FTD** The Administrators can deploy intrusion policy with intrusion rules to any interface. An interface, however, can only have one policy applied to that interface. The Administrators can also import vendor-defined signatures from Cisco, create their own intrusion rules, create rules to define which traffic is inspected and analyzed, enable anomaly rules/detections, modify thresholds and threshold duration, and configure white-list/black-list. The IPS Analysts (Intrusion Admins) Administrators can create, modify, or delete

TOE SFRs	How the SFR is Satisfied
	intrusion policies but only the IPS Administrators can deploy the policies. Here are the security roles in addition to the all-powerful "Administrator" role. • "IPS Administrator" (or Administrator): Have all privileges and access • "IPS Analyst" (or Intrusion Admin): Have all access to intrusion policies, IPS policies and network analysis privileges but cannot deploy policies • Access Admin: Have all access to access control policies but cannot deploy policies • Discovery Admin: Have all access to network discovery, application detection, and correlation features but cannot deploy policies • Security Analyst: Have all access to security event analysis feature
IPS_ABD_EXT.1[IPS] IPS_IPB_EXT.1[IPS] IPS_NTA_EXT.1[IPS] IPS_SBD_EXT.1[IPS]	**FTD Only** The TOE provides network analysis and intrusion policies as part of the FTD's intrusion detection and prevention system. The term "intrusion detection" generally refers to the process of passively analyzing network traffic for potential intrusions and storing attack data for security analysis. The term "intrusion prevention" includes the concept of intrusion detection but adds the ability to block or alter malicious traffic as it travels across the network. In an intrusion detection/prevention deployment, the TOE examines packets as such: • A network analysis policy governs how traffic is decoded and preprocessed so it can be further evaluated, especially for anomalous traffic that might signal an intrusion attempt. • An intrusion policy uses intrusion and preprocessor rules (sometimes referred to collectively as intrusion rules) to examine the decoded packets for attacks based on patterns or signatures. Without decoding and preprocessing, the TOE could not appropriately evaluate traffic for intrusions because protocol differences would make pattern matching impossible. Network analysis policies govern the traffic-handling tasks: 1. Security Intelligence uses reputation intelligence to quickly block connections to or from IP addresses, URLs, and domain names and is an early phase of access control. 2. Before traffic can be inspected by intrusion policies Security Intelligence *lists* and *feeds* are collections of IP addresses, domain names, and URLs that you can use to quickly filter traffic that matches an entry on a list or feed. • A list is a static collection that can be managed manually. • A feed is a dynamic collection that updates on an interval. Security Intelligence lists/feeds are grouped into: • DNS (Domain names) • Network (IP addresses) • URLs Predefined global Block lists and Allow lists for domains (DNS), IP addresses (Networks), and URLs are available by default and the Administrators can build on the list. Block List/Do Not

TOE SFRs	How the SFR is Satisfied
	Block List options are available on IP address, URL, and DNS requests. Using these rules to block or allow an item adds the item to the appropriate default Global list. By default, Access control and DNS policies use these Global lists. These lists can be applied on a per-policy basis. When a packet matches Block list, the packet is dropped and the TOE crafts and sends a TCP Reset packet to the original destination IP address. A network analysis policy governs packet processing in phases. First the system decodes packets through the first three TCP/IP layers, then continues with normalizing, preprocessing, and detecting protocol anomalies: • The packet decoder converts packet headers and payloads into a format that can be easily used by the preprocessors and later, intrusion rules. Each layer of the TCP/IP stack is decoded in turn, beginning with the data link layer and continuing through the network and transport layers. The packet decoder also detects various anomalous behaviors in packet headers. • The inline normalization preprocessor reformats (i.e., normalizes) traffic to minimize the chances of attackers evading detection. It prepares packets for examination by other preprocessors and intrusion rules and helps ensure that the packets the system processes are the same as the packets received by the hosts on your network. • Various network and transport layers preprocessors detect attacks that exploit IP fragmentation, perform checksum validation, and perform TCP and UDP session preprocessing. o Various application-layer protocol decoders normalize specific types of packet data into formats that the intrusion rules engine can analyze. Normalizing application-layer protocol encodings allows the system to effectively apply the same content-related intrusion rules to packets whose data is represented differently, and to obtain meaningful results. Conformance to protocols has been verified via compliance testing. o The Modbus and DNP3 SCADA preprocessors detect traffic anomalies and provide data to intrusion rules. The operations associated with the anomaly-based IPS policies are *allow the traffic flow* for any sensor interface in any mode and *allow the traffic flow* and *block/drop the traffic flow* in inline mode. Administrators can define strings to match URLs/URIs, and web page content for pattern-matching. o Several preprocessors allow administrators to detect specific threats, such as IP/TCP/UDP/ICMP port scans, ICMP/TCP flooding, DoS attacks and other rate-based attacks ("frequency"). The administrator can configure threshold[4] that mimics normal expected frequency and configure the TOE to detect and drop events exceeding the configured thresholds. When the system identifies a possible intrusion, it generates an intrusion or preprocessor event (sometimes collectively called intrusion events). Managed Sensors transmit their events to the Firepower Management Center, where the administrators can view the aggregated data and gain a greater understanding of the attacks against their network

[4] Although the term "threshold" is used in the TSS, the TOE's definition of "threshold" matches the definition of frequency in the mod_ips_v1.0. Therefore, "frequency", rather than "threshold" has been selected in the IPS_ABD_EXT.1.1 requirement.

TOE SFRs	How the SFR is Satisfied
	assets. In an inline deployment, managed Sensors can also drop packets that are known to be harmful. Each intrusion event in the database includes an event header and contains information about the event name and classification; the source and destination IP addresses; ports; the process that generated the event; and the date and time of the event, as well as contextual information about the source of the attack and its target. For packet-based events, the TOE also logs a copy of the decoded packet header and payload for the packet or packets that triggered the event. The packet decoder, the preprocessors, and the intrusion rules engine can all cause the TOE to generate an event. For examples, • If the packet decoder (configured in the network analysis policy) receives an IP packet that is less than 20 bytes, which is the size of an IP datagram without any options or payload, the decoder interprets this as anomalous traffic. If, later, the accompanying decoder rule in the intrusion policy that examines the packet is enabled, the system generates a preprocessor event. • If the IP defragmentation preprocessor encounters a series of overlapping IP fragments, the preprocessor interprets this as a possible attack and, when the accompanying preprocessor rule is enabled, the system generates a preprocessor event. • Within the intrusion rules engine, most intrusion rules are written so that they generate intrusion events when triggered by packets. Please see section 7.1 for more details on Snort rule. Until the administrator deploy new policies to the network interface, rules in the currently deployed intrusion policies behave as follows: • Disabled rules remain disabled. • Rules set to **Generate Events** continue to generate events when triggered. • Rules set to **Drop and Generate Events** continue to generate events and drop offending packets when triggered. The administrator can set thresholds for individual rules, per intrusion policy, to limit the number of times the system logs and displays an intrusion event based on how many times the event is generated within a specified time period. This can prevent the TOE from being overwhelmed with a large number of identical events. The TOE can also be configured to use intrusion rules to detect various attacks such as Teardrop, Bonk, Ping of Death, etc. The administrators can use pre-defined rule or create custom rule to detect these attacks and many more. Please reference the CC Supplemental User Guide for more details. The administrator can configure the Sensor in either a passive or inline deployment. In a passive (promiscuous) IPS deployment, the Sensor monitors traffic flowing across a network using a switch SPAN or mirror port. The SPAN or mirror port allows for traffic to be copied from other ports on the switch. This provides the system visibility within the network without being in the flow of network traffic. When configured in a passive deployment, the system cannot take certain actions such as blocking or shaping traffic. The administrator can configure one or more physical ports (Gigabit ethernet interfaces) on a managed Sensor as passive interfaces and deploy the intrusion policy to that interface via security zone (i.e., the interface is added to the zone). In an inline IPS deployment, the administrator configures

TOE SFRs	How the SFR is Satisfied
	the Sensor transparently on a network segment by binding two ports together. The administrator can configure one or more physical ports (Gigabit ethernet interfaces) on a managed Sensor as inline interfaces then assign a pair of inline interfaces to an inline set. The intrusion policy is then deployed to that inline set via security zone. The management interface (typically eth0) is separate from the other data monitoring interfaces (used as passive or inline) on the Sensor. It is used to set up and register the Sensor to the FMC.
Security Functional Requirements Drawn from mod_cpp_fw_v1.4e	
FDP_RIP.2[FW]	**FTD Only** The TOE ensures that packets transmitted through the TOE do not contain residual information from previous packets. Packets that are not the required length (for the temporary memory storage location, or for the minimum transmission unit size on the egress interface) use zeros for padding so residual data is never transmitted from the TOE. Packet handling within memory buffers ensures new packets cannot contain portions of previous packets by ensuring that when packets are written to memory locations those memory locations are padded with zeros as necessary to fill the allocated memory size, so no residual data exists within that memory range when the packet is read for transmission. This applies to data plane traffic and even administrative session traffic.
FFW_RUL_EXT.1.1[FW] FFW_RUL_EXT.1.2[FW]	**FTD Only** The TOE provides stateful traffic filtering of IPv4 and IPv6 network traffic. Administratively-defined traffic filter rules (access-lists or **Objects > Object Management > Access Control Lists > Extended**) can be applied to any interface to filter traffic based on IP parameters including source and destination address, transport layer protocol, type and code, TCP and UDP port numbers. The TOE allows establishment of communications between remote endpoints, and tracks the state of each session (e.g. initiating, established, and tear-down), and will clear established sessions after proper tear-down is completed as defined by each protocol, or when session timeouts are reached. To track the statefulness of sessions to/from and through the firewall, the TOE maintains a table of connections in various connection states and connection flags. The TOE updates the table (adding, and removing connections, and modifying states as appropriate) based on configurable connection timeout limits, and by inspecting fields within the packet headers. For further explanation of connection states, see section 7.2. The proper session establishment and termination followed by the TOE is as defined in the following RFCs: • RFC 792 (ICMPv4) • RFC 4443 (ICMPv6) • RFC 791 (IPv4) • RFC 8200 (IPv6) • TCP, RFC 793, section 2.7 Connection Establishment and Clearing • UDP, RFC 768 (not applicable, UDP is a “stateless” protocol) During initialization/startup (while the TOE is booting) the configuration has yet to be loaded, and no traffic can flow through any of its interfaces. No traffic can flow through the TOE interfaces until the POST has completed, and the configuration has been loaded. If any aspect of the POST fails during boot, the TOE will reload without forwarding traffic. If a

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td></td><td>critical component of the TOE, such as the clock or cryptographic modules, fails while the TOE is in an operational state, the TOE will reload, which stops the flow of traffic. If a component such as a network interface, which is not critical to the operation of the TOE, but may be critical to one or more traffic flows, fails while the TOE is operational, the TOE will continue to function, though all traffic flows through the failed network interface(s) will be dropped.

When traffic exceeds the maximum rate the TOE can handle, the TOE drops the excess traffic and ensures that no traffic that wouldn't pass stateful traffic filtering rules would be passed through.</td></tr>
<tr><td>FFW_RUL_EXT.1.2[FW]</td><td><u>FTD Only</u>

The TOE supports filtering of the following protocols and enforces proper session establishment, management, and termination as defined in each protocol's RFC , using the following filtering options configured via FMC:

To filter ICMPv4 or ICMPv6 Type and Code:
Policies > Access Control > Access Control > Add Rule ><ol type="a">
Zones (mapped to interfaces) > Available Zones > click either “Add to Source” or “Add to Destination”
Networks > select IPv4 networks> add to source and/or destination
Ports > Selected Destination Ports > Protocol > ICMP > select type and code
To filter ICMPv6 Type and Code: As explained above for ICMPv4, but under “Networks” select IPv6 addresses.

To filter IPv4 Source address, Destination Address, and Transport Layer Protocol:
Policies > Access Control > Access Control > Add Rule ><ol type="a">
Zones (mapped to interfaces) > Available Zones > click either “Add to Source” or “Add to Destination”
Networks > select IPv4 networks > add to source and/or destination
Ports > select a pre-named port, or create a new named protocol+port > add to source and/or destination
To filter IPv6 Source Address, Destination Address, and Transport Layer Protocol: As explained above for IPv4, but under “Networks” select IPv6 addresses.

To filter TCP Source Port and/or Destination Port: As explained above for IPv4 or IPv6, and under “Ports” select “TCP” and a port under either or both of “Selected Source Ports” and/or “Selected Destination Ports.”

To filter UDP Source Port and/or Destination Port: As explained above for IPv4 or IPv6, and under “Ports” select “UDP” and a port under either or both of “Selected Source Ports” and/or “Selected Destination Ports.”
Addresses, type of service, fragmentation data, size and padding, and IP options including loose source routing, strict source routing, and record route as defined in RFC 791 (IPv4), and RFC 8200 (IPv6);
Port numbers, sequence and acknowledgement numbers, size and padding, and control bits such as SYN, ACK, FIN, and RST as defined in RFC 793 (TCP);</td></tr>
</table>

TOE SFRs	How the SFR is Satisfied
	• Port numbers, and length as defined in RFC 768 (UDP); and • Session identifiers, sequence numbers, types, and codes as defined in RFC 792 (ICMPv4), and RFC 4443 (ICMPv6). Cisco confirms proper implementation of the RFCs through interoperability testing with Cisco and 3rd party products and through protocol compliant testing. The TOE can also support deeper packet inspection and enforce additional RFC compliance beyond session management, but such traffic inspection functionality is not defined within the NDcPP and is therefore beyond the scope of this CC certification.

FFW_RUL_EXT.1.3[FW],
FFW_RUL_EXT.1.4[FW]
	FTD Only
	
	Each traffic flow control rule on the TOE is defined as either a "permit" rule, or a "deny" rule, and any rule can also contain the keyword "log" which will cause a log message to be generated when a new session is established because it matched the rule. The TOE can be configured to generate a log message for the session establishment or an attempt at session establishment of any permitted or denied traffic. When a rule is created to explicitly allow a protocol which is implicitly allowed to spawn additional sessions, the establishment of spawned sessions is logged as well.
	
	Access Control Lists (ACLs) are only enforced after they've been applied to a network interface. Any network interface can have an ACL applied to it. Interfaces can be referred to by their identifier (e.g. GigabitEthernet 0/1).
	
	The interface types that can be assigned to an interface are:
	
	• Physical interfaces
	  o Ethernet
	  o GigabitEthernet
	  o TenGigabitEthernet
	  o Management
	• Port-channel interfaces (designated by a port-channel number)
	• Subinterface (designated by the subinterface number)
	
	The default state of an interface depends on the type and the context mode:
	
	• For the "system" context in single mode or multiple context mode, interfaces have the following default states:
	  o Physical interfaces = Disabled
	  o Subinterfaces = Enabled. However, for traffic to pass through the subinterface, the physical interface must also be enabled.
	• For any non-system context (in multiple context mode): All allocated interfaces (allocated to the context by the system context) are enabled by default, no matter what the state of the interface is in the system context. However, for traffic to pass through the interface, the interface also has to be enabled in the system context. If you shut down an interface in the system context, then that interface is down in all contexts to which that interface has been allocated.
	
	In interface configuration mode, the administrator can configure hardware settings (for physical interfaces), assign a name, assign a VLAN, assign an IP address, and configure many other settings, depending on the type of interface and the security context mode.

TOE SFRs	How the SFR is Satisfied
	For an enabled interface to pass traffic, the following interface configuration mode commands must be used (in addition to explicitly permitting traffic flow by applying and access-group to the interface): "**nameif"**, and, for routed mode, "**ip address"**. For subinterfaces, also configure the "**vlan"** command.
FFW_RUL_EXT.1.5[FW]	**FTD Only** All traffic that goes through the TOE is inspected using the Adaptive Security Algorithm and either is allowed through or dropped. A simple packet filter can check for the correct source address, destination address, and ports, but it does not check that the packet sequence or flags are correct. A filter also checks every packet against the filter, which can be a slow process. A stateful firewall like the TOE, however, takes into consideration the state of a packet: • Is this a new connection? If it is a new connection, the TOE has to check the packet against access control lists and perform other tasks to determine if the packet is allowed or denied. To perform this check, the first packet of the session goes through the "session management path," and depending on the type of traffic, it might also pass through the "control plane path." The session management path is responsible for the following tasks: – Performing the access list checks – Performing route lookups – Allocating NAT translations (xlates) – Establishing sessions in the "fast path" The TOE creates forward and reverse flows in the fast path for TCP traffic; the TOE also creates connection state information for connectionless protocols like UDP, so that they can also use the fast path. • Is this an established connection? If the connection is already established, the TOE does not need to re-check packets against the ACL; matching packets can go through the "fast" path based on attributes identified in FFW_RUL_EXT.1.5. The fast path is responsible for the following tasks: – IP checksum verification – Session lookup – TCP sequence number check – NAT translations based on existing sessions – Layer 3 and Layer 4 header adjustments Existing traffic flows are removed from the set of established traffic flows when the session inactivity timeout hits or the completion of the expected information flow.

TOE SFRs	How the SFR is Satisfied

FFW_RUL_EXT.1.6[FW],
FFW_RUL_EXT.1.7[FW]
	<u>FTD Only</u>
	
	The TOE can be configured to implement default denial of various mal-formed packets/fragments, and other illegitimate network traffic, and can be configured to log that such packets/frames were dropped.
	
	The following traffic will be denied/dropped by the TOE, and when auditing of such actions has been enabled by an administrator the TOE will generate audit messages when the action occurs:
	
	a) Packets which are invalid fragments (The TOE will count the number packets that were dropped because the packets included invalid fragments. Invalid fragments include: overlapping fragments ('teardrop' attack); and invalid IP fragment size ('ping of death' attack)).
	
	b) Fragments that cannot be completely re-assembled (The TOE will count the number of packets that fail to be reassembled. Packets that fail to be reassembled include those that exceed any of the thresholds (configured globally, or per-interface) for fragment reassembly, including limits for: the maximum number of fragments allowed for a single packet (chain size); the maximum number of fragments the TOE will hold in its IP reassembly database waiting for reassembly (size limit); and the maximum number of seconds to wait for all fragments of a packet to be received (timeout limit).)
	
	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) The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address "reserved for future use" (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
	
	g) The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;
	
	h) Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified
	
	i) Other packets defined in FFW_RUL_EXT.1.6:
	• In routed mode when the TOE receives a through-the-box:
	    o L2 broadcast packet (MAC address FF:FF:FF:FF:FF:FF)
	    o IPv4 packet with destination IP address equal to 0.0.0.0
	    o IPv4 packet with source IP address equal to 0.0.0.0
	• In routed or transparent mode when the TOE receives a through-the-box IPv4 packet with any of:
	    o first octet of the source IP address equal to zero
	    o network part of the source IP address equal to all 0's
	    o network part of the source IP address equal to all 1's
	    o source IP address host part equal to all 0's or all 1's
	    o source IP address and destination IP address are the same ("land c" attack)

TOE SFRs	How the SFR is Satisfied
	• LAND Attack – Network packets with IP source address the same as the destination IP and the destination port the same as the source port. • ICMP Error Inspect and ICMPv6 Error Inspect (ICMP error packets when the ICMP error messages are not related to any session already established in the TOE). • ICMPv6 condition (when the appliance is not able to find any established connection related to the frame embedded in the ICMPv6 error message). • ICMP Inspect bad icmp code (when an ICMP echo request/reply packet was received with a malformed code(non-zero)). The following traffic will be denied/dropped by the TOE by the default action (deny/drop) of each Access Control Policy, and if logging is enabled for the default action the TOE will generate audit messages when the action occurs: a) Packets where the source address is equal to the address of the network interface where the network packet was received. b) Packets where the source or destination address of the network packet is a link-local address. c) Packets where the source address does not belong to the networks associated with the network interface where the network packet was received, including a description of how the TOE determines whether a source address belongs to a network associated with a given network interface.
FFW_RUL_EXT.1.8[FW]	**FTD Only** The TOE administrators have control over the sequencing of access control entries (ACEs) within an access control list (ACL) to be able to set the sequence in which ACEs are applied within any ACL. The entries within an ACL are always applied in a top-down sequence, and the first entry that matches the traffic is the one that's applied, regardless of whether there may be a more precise match for the traffic further down in the ACL. By changing the ordering/numbering of entries within an ACL, the administrator chances the sequence in which the entries are compared to network traffic flows.
FFW_RUL_EXT.1.9[FW]	**FTD Only** An implicit "deny-all" rule is applied to all interfaces to which any traffic filtering rule has been applied. The implicit deny-all rule is executed after all admin-defined rules have been executed and will result in dropping all traffic that has not been explicitly permitted, or explicitly denied. If an administrator wants to log all denied traffic, a rule entry should be added that denies all traffic and logs it, e.g. "access-list sample-acl deny ip any any log".
FFW_RUL_EXT.1.10[FW]	**FTD Only** The TOE administrators can configure the maximum number of half-open TCP connections allowed by configuring a Network Analysis Policy to include SYN Attack Prevention with desired limits. After the configured limit is reached, the TOE will act as a proxy for the server and generates a SYN-ACK response to new client SYN request. When the TOE receives an ACK back from the client, it can then authenticate that the client is real and allow the connection to the server. If an ACK is not received in the configurable time frame,

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td></td><td>the session is closed, resource is returned to the free pool, and it will be counted. The default idle time until a TCP half-open connection closes is 10 minutes.</td></tr>
<tr><td>FFW_RUL_EXT.2[FW]</td><td><u>FTD Only</u>

The TOE supports TCP and UDP protocols that require dynamic establishment of secondary network sessions like FTP and the establishment of the sessions along with the dynamical definition of the rule are treated as auditable events. The TOE will manage establishment and teardown of the following protocols in accordance with the RFC for each protocol:

• FTP (File Transfer Protocol) is a TCP protocol supported in either active or passive mode:
o In active mode the client initiates the control session, and the server initiates the data session to a client port provided by the client;
o For active FTP to be allowed through the TOE, the firewall rules must explicitly permit the control session from the client to the server, and “inspect ftp” must be enabled. The TOE will then explicitly permit a control session to be initiated from the client to the server, and implicitly permit data sessions to be initiated from the server to the client while the control session is active.
o In passive (PASV) mode, the client initiates the control session, and the client also initiates the data session to a secondary port provided to the client by the server.

For passive FTP to be permitted through the TOE, the firewall rules must explicitly permit the control session from the client to the server, and “inspect ftp” must be enabled with the “match passive-ftp” option enabled. That feature will cause the TOE to look for the PASV or EPSV commands in the FTP control traffic and for the server’s destination port, and dynamically permit the data session.</td></tr>
<tr><td colspan="2">Security Functional Requirements Drawn from mod_vpngw_v1.3</td></tr>
<tr><td>FCS_CKM.1/IKE [VPN]</td><td>See FCS_CKM.1</td></tr>
<tr><td>FPF_RUL_EXT.1 [VPN]</td><td><u>FTD Only</u>

An authorized administrator can define the traffic that needs to be protected by FTD by configuring access lists (permit, deny, log), applying these access lists to VPN policies (peer-to-peer and remote access VPN policies), and then assigning those policies to FTDs. Therefore, traffic may be selected on the basis of the source and destination address, and optionally the Layer 4 protocol and port.

The TOE enforces information flow policies on network packets that are received by TOE interfaces and leave the TOE through other TOE interfaces. When network packets are received on a TOE interface, the TOE verifies whether the network traffic is allowed or not and performs one of the following actions, pass/not pass information, as well as optional logging.

The TOE implements rules that define the permitted flow of traffic between interfaces of the TOE for unauthenticated traffic. These rules control whether a packet is transferred from one interface to another based on:

1. Presumed address of source</td></tr>
</table>

TOE SFRs	How the SFR is Satisfied
	2. Presumed address of destination 3. Transport layer protocol (or next header in IPv6) 4. Service used (UDP or TCP ports, both source and destination) 5. Network interface on which the connection request occurs These rules are supported for the following protocols: RFC 791(IPv4); RFC 2460 (IPv6); RFC 793 (TCP); RFC 768 (UDP). TOE compliance with these protocols is verified via regular quality assurance, regression, and interoperability testing. The TOE supports all IPv4 protocols excluding Protocol 2 (IGMP) which is not routable and thus will not be forwarded by the TOE. The TOE supports the following 15 IPv6 protocols: • Transport Layer Protocol 4 - IPv4 encapsulation • Transport Layer Protocol 6 - Transmission Control • Transport Layer Protocol 8 - Exterior Gateway Protocol • Transport Layer Protocol 9 - any private interior gateway • Transport Layer Protocol 17 - User Datagram • Transport Layer Protocol 41 - IPv6 encapsulation • Transport Layer Protocol 46 - Reservation Protocol • Transport Layer Protocol 47 - General Routing Encapsulation • Transport Layer Protocol 49 - BNA • Transport Layer Protocol 58 - ICMP for IPv6 • Transport Layer Protocol 59 - No Next Header for IPv6 • Transport Layer Protocol 88 - TCF • Transport Layer Protocol 89 - EIGRP • Transport Layer Protocol 105 - SCPS Transport Layer Protocol • Transport Layer Protocol 112 - Virtual Router Redundancy Protocol All other IPv6 protocols from the RFC Values for IPv4 and IPv6 table in the MOD VPNGW SD v1.1 are dropped by default by the TOE. Packets will be dropped unless a specific rule has been set up to allow the packet to pass (where the attributes of the packet match the attributes in the rule and the action associated with the rule is to pass traffic). Rules are enforced on a first match basis from the top down. As soon as a match is found the action associated with the rule is applied. These rules are entered in the form of access lists at the CLI (via ‘access list’ and ‘access group’ commands). These interfaces reject traffic when the traffic arrives on an external TOE interface, and the source address is an external IT entity on an internal network; These interfaces reject traffic when the traffic arrives on an internal TOE interface, and the source address is an external IT entity on the external network;

TOE SFRs	How the SFR is Satisfied
	These interfaces reject traffic when the traffic arrives on either an internal or external TOE interface, and the source address is an external IT entity on a broadcast network; These interfaces reject traffic when the traffic arrives on either an internal or external TOE interface, and the source address is an external IT entity on the loopback network; These interfaces reject requests in which the subject specifies the route for information to flow when it is in route to its destination; and For application protocols supported by the TOE (e.g., DNS, HTTP, SMTP, and POP3), these interfaces deny any access or service requests that do not conform to its associated published protocol specification (e.g., RFC). This is accomplished through protocol filtering proxies that are designed for that purpose. Otherwise, these interfaces pass traffic only when its source address matches the network interface originating the traffic to the network interface corresponding to the traffic's destination address. During the boot cycle, the TOE first powers on hardware, loads the image, and executes the power on self-tests. Until the power on self tests successfully complete, the interfaces to the TOE are deactivated. Once the tests complete, the interfaces become active and the rules associated with the interface become immediately operational. There is no state during initialization/ startup that the access lists are not enforced on an interface. During initialization/startup (while the TOE is booting) the configuration has yet to be loaded, and no traffic can flow through any of its interfaces. No traffic can flow through the TOE interfaces until the POST has completed, and the configuration has been loaded. If any aspect of the POST fails during boot, the TOE will reload without forwarding traffic. If a critical component of the TOE, such as the clock or cryptographic modules, fails while the TOE is in an operational state, the TOE will reload, which stops the flow of traffic. If a component such as a network interface, which is not critical to the operation of the TOE, but may be critical to one or more traffic flows, fails while the TOE is operational, the TOE will continue to function, though all traffic flows through the failed network interface(s) will be dropped.
FPT_FLS.1/SelfTest[VPN]	**FTD Only** Noise source health tests are run both periodically and at start-up to determine the functional health of the noise source. These tests are specifically designed to catch catastrophic losses in the overall entropy associated with the noise source. Tests are run on the raw noise output, before the application of any conditioners. If a noise source fails the health test either at start-up or after the device is operational, the platform will be shut down. Whenever a failure (e.g., POST or integrity test fails) occurs within the FTD that results in the FTD ceasing operation, the FTD securely disables its interfaces to prevent the unintentional flow of any information to or from the FTD and reloads. So long as the failures persist, the FTD will continue to reload. This functionally prevents any failure from causing an unauthorized information flow. There are no failures that circumvent this protection.
FPT_TST_EXT.3[VPN]	See FPT_TST_EXT.1

TOE SFRs	How the SFR is Satisfied
FTA_SSL.3/VPN[VPN]	**FTD** When a remote VPN client session reaches a period of inactivity, its connection is terminated, and it must re-establish the connection with new authentication to resume operation. This period of inactivity is set by the administrator using **Objects > Object Management > Group Policy > Session Settings > Idle Timeout** in the VPN configuration. The Group Policy is then tied to a Connection Profile.
FTA_TSE.1[VPN]	**FTD** The TOE allows for creation of ACLs that restrict VPN connectivity-based client's IP address (location). These ACLs allow customization of all of these properties to allow or deny access (**Objects > Object Management > Group Policy > Traffic Filter Fields > Access List Filter)**. In addition, the administrator can create Group Policy tied to Connection Profile (**Objects > Object Management > Group Policy > Session Settings > Access Hours)** which can be used to restrict access based on date and time.
FTA_VCM_EXT.1 [VPN]	**FTD** The TOE provides the option to assign the remotely connecting VPN client an internal network IP address. The **Objects > Object Management > Address Pools** can be used to define the range of IP and IPv6 addresses to be available for use.
FTP_ITC.1.1[VPN]	See FTP_ITC.1

7 SUPPLEMENTAL TOE SUMMARY SPECIFICATION INFORMATION

7.1 Intrusion Rule Definition

An intrusion rule is a set of keywords and arguments that the system uses to detect attempts to exploit vulnerabilities on your network. As the system analyzes network traffic, it compares packets against the conditions specified in each rule. If the packet data matches all the conditions specified in a rule, the rule triggers. If a rule is an alert rule, it generates an intrusion event. If it is a pass rule, it ignores the traffic. For a drop rule in an inline deployment, the system drops the packet and generates an event. The administrator can view and evaluate intrusion events from the FMC web interface.

All rules contain two logical sections: the rule header and the rule options. The rule header contains:

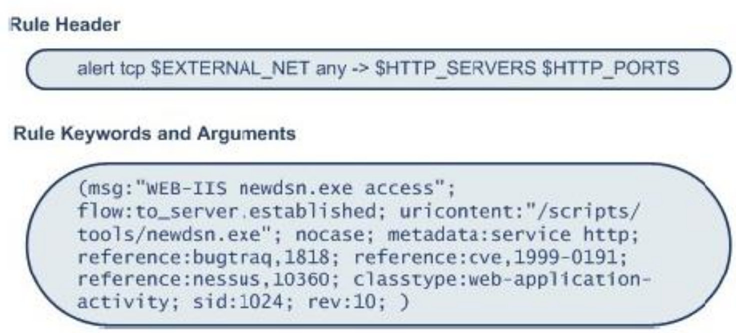
- the rule's action or type
- the protocol
- the source and destination IP addresses and netmasks
- direction indicators showing the flow of traffic from source to destination
- the source and destination ports

The rule options section contains:

- event messages
- keywords and their parameters and arguments
- patterns that a packet's payload must match to trigger the rule
- specifications of which parts of the packet the rules engine should inspect

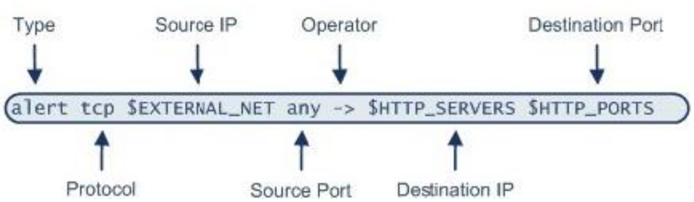
The following diagram illustrates the parts of a rule:

For example,



7.1.1 Intrusion Rule Header

Every rule has a rule header containing parameters and arguments. The following illustrates parts of a rule header:



Action (*alert*) – generates an intrusion event when triggered and operations (allow, block/drop) associated with policies.
Protocol (*tcp*) – Tests TCP traffic only. ICMPv4, ICMPv6, IPv4, IPv6, TCP, and UDP protocols are supported.
Source IP (*$EXTERNAL_NET*) – Tests traffic coming from any host that is not on your internal network.
Source Port (*any*) – Tests traffic coming from any port on the originating host.
Operate (->) – Tests external traffic destined for the web servers on your network.
Destination IP (*$HTTP_SERVERS*) - Tests traffic to be delivered to any host specified as a web server on your internal network. Both IP and IPv6 addresses and ranges are supported.
Destination Port (*$HTTP_PORTS*) - Tests traffic delivered to an HTTP port on your internal network.

7.1.2 Intrusion Rule Options and Keywords

Rule options follow the rule header and are enclosed inside a pair of parentheses. There may be one option or many and the options are separated with a semicolon. If you use multiple options, these options form a logical AND. The action in the rule header is invoked only when all criteria in the options are true. In general, an option may have two parts: a keyword and an argument.

The *message* keyword: Specify meaningful text that appears as a message when the rule triggers.

The *ack* keyword: Specify the acknowledgement value. For example, (flags: A; ack: 0; msg: "TCP ping detected";)means receive a TCP packet with the A flag set and the acknowledgement contains a value of 0.

The *content* keyword: Specify data pattern inside a packet. The pattern may be presented in the form of an ASCII string or as binary data in the form of hexadecimal characters.

The *offset* keyword: Specify a certain offset from the start of the data part of the packet to search.

The *dsize* keyword: Specify the length of the data part of a packet.

The *flags* keyword: Find out which flag bits are set inside the TCP header of a packet.

The *fragbits* keyword: Find out which three frag bits (Reserved, Don't Frag, More Frag) in the IP headers.

The *fragoffset* keyword: Tests the offset of a fragmented packet.

The *itype* keyword: Specify the ICMP type.

The *icode* keyword: Specify the ICMP code.

The *ipopts* keyword: Specify the IP Options. Record Route, Loose Source Routing, Strict Source Routing.

The *ip_proto* keyword: Specify the IP protocol number.

The *id* keyword: Specify the IP header fragment identification field

The *nocase* keyword: Its only purpose is to make a case insensitive search of a pattern within the data part of a packet. It is used in conjunction with the *content* keyword.

The *seq* keyword: Specify the sequence number of a TCP packet.

The *window* keyword: Specify the TCP window size.

The *flow* keyword: Apply a rule on TCP sessions to packets flowing in a particular direction.

The *tos* keyword: Detect a specific value in the Type of Service (TOS) field of the IP header.

The ttl keyword: Detect Time to Live value in the IP header of the packet.

7.2 Tracking of Stateful Firewall Connections

7.2.1 Establishment and Maintenance of Stateful Connections

As network traffic enters an interface of the TOE, the TOE inspects the packet header information to determine whether the packet is allowed by access control lists, and whether an established connection already exists for that specific traffic flow. The TOE maintains and continuously updates connection state tables to keep tracked of establishment, teardown, and open sessions. To help determine whether a packet can be part of a new session or an established session, the TOE uses information in the packet header and protocol header fields to determine the session state to which the packet applies as defined by the RFC for each protocol.

7.2.2 Viewing Connections and Connection States

To display the connection state for the designated connection type, use the **show conn** command in privileged EXEC mode. This command supports IPv4 and IPv6 addresses. The syntax is:

show conn [**count** | [**all**] [**detail**] [**long**] [**state** *state_type*] [**protocol** {**tcp** | **udp**}] [**scansafe**] [**address** *src_ip*[-*src_ip*] [**netmask** *mask*]] [**port** *src_port*[-*src_port*]] [**address** *dest_ip*[-*dest_ip*] [**netmask** *mask*]] [**port** *dest_port*[-*dest_port*]] [**user-identity** | **user** [*domain_nickname*\\]*user_name* | **user-group** [*domain_nickname*\\\\]*user_group_name*] | **security-group**]

The **show conn** command displays the number of active TCP and UDP connections, and provides information about connections of various types. By default, the output of "**show conn**" shows only the through-the-TOE connections. To include connections to/from the TOE itself in the command output, add the **all** keyword, "**show conn all**".

Table 26: Syntax Description

address	(Optional) Displays connections with the specified source or destination IP address.
all	(Optional) Displays connections that are to the device or from the device, in addition to through-traffic connections.
count	(Optional) Displays the number of active connections.

dest_ip	(Optional) Specifies the destination IP address (IPv4 or IPv6). To specify a range, separate the IP addresses with a dash (-). For example: 10.1.1.1-10.1.1.5
dest_port	(Optional) Specifies the destination port number. To specify a range, separate the port numbers with a dash (-). For example: 1000-2000
detail	(Optional) Displays connections in detail, including translation type and interface information.
long	(Optional) Displays connections in long format.
netmask *mask*	(Optional) Specifies a subnet mask for use with the given IP address.
port	(Optional) Displays connections with the specified source or destination port.
protocol {**tcp | udp**}	(Optional) Specifies the connection protocol, which can be **tcp** or **udp**.
scansafe	(Optional) Shows connections being forwarded to the Cloud Web Security server.
security-group	(Optional) Specifies that all connections displayed belong to the specified security group.
src_ip	(Optional) Specifies the source IP address (IPv4 or IPv6). To specify a range, separate the IP addresses with a dash (-). For example: 10.1.1.1-10.1.1.5
src_port	(Optional) Specifies the source port number. To specify a range, separate the port numbers with a dash (-). For example: 1000-2000
state *state_type*	(Optional) Specifies the connection state type. See Table 27 for a list of the keywords available for connection state types.
user [*domain_nickname*\\] *user_name*	(Optional) Specifies that all connections displayed belong to the specified user. When you do not include the *domain_nickname* argument, the TOE displays information for the user in the default domain.
user-group [*domain_nickname*\\\\] *user_group_name*	(Optional) Specifies that all connections displayed belong to the specified user group. When you do not include the *domain_nickname* argument, the TOE displays information for the user group in the default domain.
user-identity	(Optional) Specifies that the TOE display all connections for the Identity Firewall feature. When displaying the connections, the TOE displays the user name and IP address when it identifies a matching user. Similarly, the TOE displays the host name and an IP address when it identifies a matching host.

The connection types that you can specify using the **show conn state** command are defined in the table below. When specifying multiple connection types, use commas without spaces to separate the keywords.

Table 27: Connection State Types

Keyword	Connection Type Displayed
up	Connections in the up state.
conn_inbound	Inbound connections.
ctiqbe	CTIQBE connections
data_in	Inbound data connections.
data_out	Outbound data connections.
finin	FIN inbound connections.
finout	FIN outbound connections.
h225	H.225 connections
h323	H.323 connections
http_get	HTTP get connections.
mgcp	MGCP connections.
nojava	Connections that deny access to Java applets.
rpc	RPC connections.
service_module	Connections being scanned by an SSM.
sip	SIP connections.
skinny	SCCP connections.
smtp_data	SMTP mail data connections.
sqlnet_fixup_data	SQL*Net data inspection engine connections.
tcp_embryonic	TCP embryonic connections.
vpn_orphan	Orphaned VPN tunneled flows.

When using the **detail** option, the TOE displays information about the translation type and interface information using the connection flags defined in the table below.

Table 28: Connection State Flags

Flag	Description
a	awaiting outside ACK to SYN
A	awaiting inside ACK to SYN
b	TCP state bypass. By default, all traffic that passes through the Cisco Adaptive Security Appliance (FTD) is inspected using the Adaptive Security Algorithm and is either allowed through or dropped based on the security policy. In order to maximize the firewall performance, the FTD checks the state of each packet (for example, is this a new connection or an established connection?) and assigns it to either the session management path (a new connection SYN packet), the fast path (an established connection), or the control plane path (advanced inspection). TCP packets that match existing connections in the fast path can pass through the adaptive security appliance without rechecking every aspect of the security policy. This feature maximizes performance.
B	initial SYN from outside
C	Computer Telephony Interface Quick Buffer Encoding (CTIQBE) media connection
d	dump
D	DNS
E	outside back connection. This is a secondary data connection that must be initiated from the inside host. For example, using FTP, after the inside client issues the PASV command and the outside server accepts, the FTD preallocates an outside back connection with this flag set. If the inside client

	attempts to connect back to the server, then the FTD denies this connection attempt. Only the outside server can use the preallocated secondary connection.
f	inside FIN
F	outside FIN
g	Media Gateway Control Protocol (MGCP) connection
G	connection is part of a group The G flag indicates the connection is part of a group. It is set by the GRE and FTP Strict fixups to designate the control connection and all its associated secondary connections. If the control connection terminates, then all associated secondary connections are also terminated.
h	H.225
H	H.323
i	incomplete TCP or UDP connection
I	inbound data
k	Skinny Client Control Protocol (SCCP) media connection
K	GTP t3-response
m	SIP media connection
M	SMTP data
O	outbound data
p	replicated (unused)
P	inside back connection This is a secondary data connection that must be initiated from the inside host. For example, using FTP, after the inside client issues the PORT command and the outside server accepts, the FTD preallocates an inside back connection with this flag set. If the outside server attempts to connect back to the client, then the FTD denies this connection attempt. Only the inside client can use the preallocated secondary connection.
q	SQL*Net data
r	inside acknowledged FIN
R	If TCP: outside acknowledged FIN for TCP connection If UDP: UDP RPC2 Because each row of “show conn” command output represents one connection (TCP or UDP), there will be only one R flag per row.
s	awaiting outside SYN
S	awaiting inside SYN
t	SIP transient connection For a UDP connection, the value t indicates that it will timeout after one minute.
T	SIP connection For UDP connections, the value T indicates that the connection will timeout according to the value specified using the “timeout sip” command.
U	up
V	VPN orphan

W	WAAS
X	Inspected by the service module, such as a CSC SSM.
y	For clustering, identifies a backup owner flow.
Y	For clustering, identifies a director flow.
z	For clustering, identifies a forwarder flow.
Z	Cloud Web Security

A single connection is created for multiple DNS sessions, as long as they are between the same two hosts, and the sessions have the same 5-tuple (source/destination IP address, source/destination port, and protocol). DNS identification is tracked by *app_id*, and the idle timer for each app_id runs independently. Because the app_id expires independently, a legitimate DNS response can only pass through the TOE within a limited period of time and there is no resource build-up. However, when the **show conn** command is entered, you will see the idle timer of a DNS connection being reset by a new DNS session. This is due to the nature of the shared DNS connection and is by design.

When the TOE creates a pinhole to allow secondary connections, this is shown as an incomplete conn by the **show conn** command. Incomplete connections will be cleared from the connections table when they reach their timeout limit, and can be cleared manually by using the "**clear conn"** command. When there is no TCP traffic for the period of inactivity defined by the **timeout conn** command (by default, 1:00:00), the connection is closed and the corresponding conn flag entries are no longer displayed.

If a LAN-to-LAN/Network-Extension Mode tunnel drops and does not come back, there might be a number of orphaned tunnel flows. These flows are not torn down as a result of the tunnel going down, but all the data attempting to flow through them is dropped. The **show conn** command output shows these orphaned flows with the **V** flag.

Table 29: TCP connection directionality flags

Flag	Description
B	Initial SYN from outside
a	Awaiting outside ACK to SYN
A	Awaiting inside ACK to SYN
f	Inside FIN
F	Outside FIN
s	Awaiting outside SYN
S	Awaiting inside SYN

7.2.3 Examples

The following is sample output from the **show conn** command. This example shows a TCP session connection from inside host 10.1.1.15 to the outside Telnet server at 10.10.49.10. Because there is no B flag, the connection is initiated from the inside. The "U", "I", and "O" flags denote that the connection is active and has received inbound and outbound data.

hostname# **show conn**

54 in use, 123 most used

TCP out 10.10.49.10:23 in 10.1.1.15:1026 idle 0:00:22, bytes 1774, flags UIO

UDP out 10.10.49.10:31649 in 10.1.1.15:1028 idle 0:00:14, bytes 0, flags D-

TCP dmz 10.10.10.50:50026 inside 192.168.1.22:5060, idle 0:00:24, bytes 1940435, flags UTIOB

TCP dmz 10.10.10.50:49764 inside 192.168.1.21:5060, idle 0:00:42, bytes 2328346, flags UTIOB

TCP dmz 10.10.10.51:50196 inside 192.168.1.22:2000, idle 0:00:04, bytes 31464, flags UIB

TCP dmz 10.10.10.51:52738 inside 192.168.1.21:2000, idle 0:00:09, bytes 129156, flags UIOB

TCP dmz 10.10.10.50:49764 inside 192.168.1.21:0, idle 0:00:42, bytes 0, flags Ti

TCP outside 192.168.1.10(20.20.20.24):49736 inside 192.168.1.21:0, idle 0:01:32, bytes 0, flags Ti

TCP dmz 10.10.10.50:50026 inside 192.168.1.22:0, idle 0:00:24, bytes 0, flags Ti

TCP outside 192.168.1.10(20.20.20.24):50663 inside 192.168.1.22:0, idle 0:01:34, bytes 0, flags Ti

TCP dmz 10.10.10.50:50026 inside 192.168.1.22:0, idle 0:02:24, bytes 0, flags Ti

TCP outside 192.168.1.10(20.20.20.24):50663 inside 192.168.1.22:0, idle 0:03:34, bytes 0, flags Ti

TCP dmz 10.10.10.50:50026 inside 192.168.1.22:0, idle 0:04:24, bytes 0, flags Ti

TCP outside 192.168.1.10(20.20.20.24):50663 inside 192.168.1.22:0, idle 0:05:34, bytes 0, flags Ti

TCP dmz 10.10.10.50:50026 inside 192.168.1.22:0, idle 0:06:24, bytes 0, flags Ti

TCP outside 192.168.1.10(20.20.20.24):50663 inside 192.168.1.22:0, idle 0:07:34, bytes 0, flags Ti

The following is sample output from the **show conn detail** command. This example shows a UDP connection from outside host 10.10.49.10 to inside host 10.1.1.15. The D flag denotes that this is a DNS connection. The number 1028 is the DNS ID over the connection.

hostname# **show conn detail**

54 in use, 123 most used

Flags: A - awaiting inside ACK to SYN, a - awaiting outside ACK to SYN,

B - initial SYN from outside, b - TCP state-bypass or nailed, C - CTIQBE media,

D - DNS, d - dump, E - outside back connection, F - outside FIN, f - inside FIN,

G - group, g - MGCP, H - H.323, h - H.225.0, I - inbound data,

i - incomplete, J - GTP, j - GTP data, K - GTP t3-response

k - Skinny media, M - SMTP data, m - SIP media, n - GUP

O - outbound data, P - inside back connection, p - Phone-proxy TFTP connection,

q - SQL*Net data, R - outside acknowledged FIN,

R - UDP SUNRPC, r - inside acknowledged FIN, S - awaiting inside SYN,

s - awaiting outside SYN, T - SIP, t - SIP transient, U - up,

V - VPN orphan, W - WAAS,

X - inspected by service module

TCP outside:10.10.49.10/23 inside:10.1.1.15/1026, flags UIO, idle 39s, uptime 1D19h, timeout 1h0m, bytes 1940435

UDP outside:10.10.49.10/31649 inside:10.1.1.15/1028, flags dD, idle 39s, uptime 1D19h, timeout 1h0m, bytes 1940435

TCP dmz:10.10.10.50/50026 inside:192.168.1.22/5060, flags UTIOB, idle 39s, uptime 1D19h, timeout 1h0m, bytes 1940435

TCP dmz:10.10.10.50/49764 inside:192.168.1.21/5060, flags UTIOB, idle 56s, uptime 1D19h, timeout 1h0m, bytes 2328346

TCP dmz:10.10.10.51/50196 inside:192.168.1.22/2000, flags UIB, idle 18s, uptime 1D19h, timeout 1h0m, bytes 31464

TCP dmz:10.10.10.51/52738 inside:192.168.1.21/2000, flags UIOB, idle 23s, uptime 1D19h, timeout 1h0m, bytes 129156

TCP outside:10.132.64.166/52510 inside:192.168.1.35/2000, flags UIOB, idle 3s, uptime 1D21h, timeout 1h0m, bytes 357405

TCP outside:10.132.64.81/5321 inside:192.168.1.22/5060, flags UTIOB, idle 1m48s, uptime 1D21h, timeout 1h0m, bytes 2083129

TCP outside:10.132.64.81/5320 inside:192.168.1.21/5060, flags UTIOB, idle 1m46s, uptime 1D21h, timeout 1h0m, bytes 2500529

TCP outside:10.132.64.81/5319 inside:192.168.1.22/2000, flags UIOB, idle 31s, uptime 1D21h, timeout 1h0m, bytes 32718

TCP outside:10.132.64.81/5315 inside:192.168.1.21/2000, flags UIOB, idle 14s, uptime 1D21h, timeout 1h0m, bytes 358694

TCP outside:10.132.64.80/52596 inside:192.168.1.22/2000, flags UIOB, idle 8s, uptime 1D21h, timeout 1h0m, bytes 32742

TCP outside:10.132.64.80/52834 inside:192.168.1.21/2000, flags UIOB, idle 6s, uptime 1D21h, timeout 1h0m, bytes 358582

TCP outside:10.132.64.167/50250 inside:192.168.1.35/2000, flags UIOB, idle 26s, uptime 1D21h, timeout 1h0m, bytes 375617

7.3 Key Zeroization

The following table describes the key zeroization referenced by FCS_CKM.4 provided by the TOE. DRAM (dynamic random access memory) is volatile memory and NVRAM (non-volatile random access memory) is non-volatile “flash” memory.

Table 30: TOE Key Zeroization

Critical Security Parameters (CSPs)	Zeroization Cause and Effect
Diffie-Hellman Shared Secret	Automatically zeroized after completion of DH exchange, by calling a specific API within the two crypto modules, when module is shutdown, or reinitialized. Storage: DRAM Overwritten with: 0x00
Diffie Hellman Private and Public Exponent	Automatically zeroized upon completion of DH exchange, by calling a specific API within the two crypto modules, and when module is shutdown, or reinitialized. Storage: DRAM Overwritten with: 0x00
skeyid	Session Encryption Key and IKE Session Authentication Key. Automatically zeroized after IKE session terminated. Storage: DRAM Overwritten with: 0x00
skeyid_d	Session Encryption Key and IKE Session Authentication Key. Automatically zeroized after IKE session terminated. Storage: DRAM Overwritten with: 0x00

Critical Security Parameters (CSPs)	Zeroization Cause and Effect
IKE Session Encryption Key	Session Encryption Key and IKE Session Authentication Key. Automatically zeroized after IKE session terminated. Storage: DRAM Overwritten with: 0x00
IKE Session Authentication Key	Session Encryption Key and IKE Session Authentication Key. Automatically zeroized after IKE session terminated. Storage: DRAM Overwritten with: 0x00
ISAKMP Preshared	Zeroized using the following command: **# no crypto isakmp key** Storage: NVRAM Overwritten with: 0x00
IKE RSA and ECDSA Private and Public Keys	Automatically overwritten when a new key is generated or zeroized using the following commands: **# crypto key zeroize rsa** **# crypto key zeroize ec** Storage: NVRAM Overwritten with: 0x00
IPsec Encryption Key	Automatically zeroized when IPsec session terminated. Storage: DRAM Overwritten with: 0x00
IPsec Authentication Key	Automatically zeroized when IPsec session terminated. Storage: DRAM Overwritten with: 0x00
SSHv2 Private and Public Key	Automatically zeroized upon generation of a new key Storage: NVRAM Overwritten with: 0x00

Critical Security Parameters (CSPs)	Zeroization Cause and Effect
SSHv2 Session Key	Automatically zeroized when the SSH session is terminated. Storage: NVRAM Overwritten with: 0x00
All CSPs	Zeroized on-demand on all file systems via the "erase" command. Storage: NVRAM Overwritten with: 0x00
TLS Server Private Key	Zeroized when the HTTPS server is no longer in use. Storage: NVRAM Overwritten with: 0x00

7.4 CAVP Certificate Equivalence

The TOE models and processors included in the evaluation are shown in the following table. The TOE includes multiple cryptographic modules across the range of TOE components. These modules are commonly referred to as FOM (FIPS Object Models). The CAVP-certified FOM of the TOE are listed in the table below (Table 31) along with the CPU for which they were certified, and the TOE component on which they're used. The table on the following page (Table 32) lists the CAVP certificate numbers for each FOM for each applicable SFR.

Table 31: Model Processors, FOM and CAVP Cert

CPU Family	CPU Model (Microarchitecture)	FOM	Physical Appliances, Modules, and Blades	CAVP Certificate #
FTD				
Intel Xeon Scalable	Intel Xeon Silver 4116 (Skylake)	CiscoSSL FOM Cryptographic Implementation 7.3a	Firepower 4112, 4115	A4446
	Intel Xeon Gold 6130 (Skylake)		Firepower 4125	
	Intel Xeon Gold 6138 (Skylake)		Firepower 9300 (SM-40)	

CPU Family	CPU Model (Microarchitecture)	FOM	Physical Appliances, Modules, and Blades	CAVP Certificate #
	Intel Xeon Gold 6152 (Skylake)		Firepower 4145	
	Intel Xeon Platinum 8160 (Skylake)		Firepower 9300 (SM-48)	
	Intel Xeon Platinum 8176 (Skylake)		Firepower 9300 (SM-56)	
FXOS				
Intel Core i3	Intel i3-3115C (Ivy Bridge)	CiscoSSL FOM Cryptographic Implementation 7.3a	Supervisor Blade (Firepower 4112, 4115, 4125 and 4145 and 9300)	A4446
Hardware Cryptographic Acceleration (for IPsec)				
	CN3550 (NITROX III)	Nitrox III series die (for ASA, for crypto acceleration)	Firepower 4112, and 9300 (SM-40)	Table 32(Column -Nitrox III series die)
	CN5560 (NITROX V)	Nitrox-V GC (for ASA, for crypto acceleration)	Firepower 4115, 4125, 4145 and 9300 (SM-48 and 56)	Table 32(Column - Nitrox-V GC)
FMC				
Intel Xeon Scalable	Intel Xeon Silver 4110 (Skylake)	CiscoSSL FOM Cryptographic Implementation 7.3a	FMC 1600, FMC 2600	A4446
	Intel Xeon Silver 4214 (Cascade Lake)		FMC 4600	
AMD EPYC	AMD EPYC 7232P (Zen 2)		FMC 1700	
	AMD EPYC 7282 (Zen 2)		FMC 2700	
	AMD EPYC 7352 (Zen 2)		FMC 4700	
FMCv				

CPU Family	CPU Model (Microarchitecture)	FOM	Physical Appliances, Modules, and Blades	CAVP Certificate #
Xeon Bronze w/ Linux 4 on ESXi 7.0	Intel Xeon Bronze 3104 (Skylake) w/ Linux 4 on ESXi 7.0	CiscoSSL FOM Virtual 7.3a	UCSC-C220-M5, UCSC-C240-M5 and UCSC-C480-M5	A4595
	Intel Xeon Bronze 3408U (Sapphire Rapids) w/ Linux 4 on ESXi 7.0		UCSC-C220-M7, UCSC-C240-M7	
Xeon Silver w/ Linux 4 on ESXi 7.0	Intel Xeon Silver 4110 (Skylake) w/ Linux 4 on ESXi 7.0		UCSC-C220-M5, UCSC-C240-M5 and UCSC-C480-M5	
	Intel Xeon Silver 4310 (Ice Lake) w/ Linux 4 on ESXi 7.0		UCSC-C220-M6, UCSC-C240-M6	
	Intel Xeon Silver 4314 (Ice Lake) w/ Linux 4 on ESXi 7.0		UCSC-C220-M6, UCSC-C240-M6	
	Intel Xeon Silver 4316 (Ice Lake) w/ Linux 4 on ESXi 7.0		UCSC-C220-M6, UCSC-C240-M6	
	Intel Xeon Silver 4410T (Sapphire Rapids) w/ Linux 4 on ESXi 7.0		UCSC-C220-M7, UCSC-C240-M7	
Xeon Gold w/ Linux 4 on ESXi 7.0	Intel Xeon® Gold 6128 (Skylake) w/ Linux 4 on ESXi 7.0		UCSC-C220-M5, UCSC-C240-M5 and UCSC-C480-M5	
	Intel Xeon Gold 5315Y (Ice Lake) w/ Linux 4 on ESXi 7.0		UCSC-C220-M6, UCSC-C240-M6	
	Intel Xeon Gold 5318N (Ice Lake) w/ Linux 4 on ESXi 7.0		UCSC-C220-M6, UCSC-C240-M6	
	Intel Xeon Gold 6312U (Ice Lake) w/ Linux 4 on ESXi 7.0		UCSC-C220-M6, UCSC-C240-M6	

CPU Family	CPU Model (Microarchitecture)	FOM	Physical Appliances, Modules, and Blades	CAVP Certificate #
	Intel Xeon Gold 6342 (Ice Lake) w/ Linux 4 on ESXi 7.0		UCSC-C220-M6, UCSC-C240-M6	
	Intel Xeon Gold 5411N (Sapphire Rapids) w/ Linux 4 on ESXi 7.0		UCSC-C220-M7, UCSC-C240-M7	
Intel Xeon Platinum w/ Linux 4 on ESXi 7.0	Intel Xeon Platinum 8153 (Skylake) w/ Linux 4 on ESXi 7.0		UCSC-C220-M5, UCSC-C240-M5 and UCSC-C480-M5	
	Intel Xeon Platinum 8160 (Skylake) w/ Linux 4 on ESXi 7.0		UCSC-C220-M5, UCSC-C240-M5 and UCSC-C480-M5	
	Intel Xeon Platinum 8351N (Ice Lake) w/ Linux 4 on ESXi 7.0		UCSC-C220-M6, UCSC-C240-M6	
	Intel Xeon Platinum 8444H (Sapphire Rapids) w/ Linux 4 on ESXi 7.0		UCSC-C220-M7, UCSC-C240-M7	
	Intel Xeon Platinum 8452Y (Sapphire Rapids) w/ Linux 4 on ESXi 7.0		UCSC-C220-M7, UCSC-C240-M7	
AMD EPYC 7002/7003 w/ Linux 4 on ESXi 7.0	AMD EPYC 7232P (Zen 2) w/ Linux 4 on ESXi 7.0		UCSC-C225-M6	
	AMD EPYC 7252 (Zen 2) w/ Linux 4 on ESXi 7.0			
	AMD EPYC 7262 (Zen 2) w/ Linux 4 on ESXi 7.0			
	AMD EPYC 7272 (Zen 2) w/ Linux 4 on ESXi 7.0			
	AMD EPYC 7282 (Zen 2) w/ Linux 4 on ESXi 7.0			
	AMD EPYC 72F3 (Zen 3) w/ Linux 4 on ESXi 7.0			

CPU Family	CPU Model (Microarchitecture)	FOM	Physical Appliances, Modules, and Blades	CAVP Certificate #
	AMD EPYC 7302 (Zen 2) w/ Linux 4 on ESXi 7.0			
	AMD EPYC 7313 (Zen 3) w/ Linux 4 on ESXi 7.0			
	AMD EPYC 7343 (Zen 3) w/ Linux 4 on ESXi 7.0			
	AMD EPYC 7352 (Zen 2) w/ Linux 4 on ESXi 7.0			
	AMD EPYC 7373X (Zen 3) w/ Linux 4 on ESXi 7.0			
	AMD EPYC 73F3 (Zen 3) w/ Linux 4 on ESXi 7.0			
	AMD EPYC 7402 (Zen 2) w/ Linux 4 on ESXi 7.0			
	AMD EPYC 74F3 (Zen 3) w/ Linux 4 on ESXi 7.0			
	AMD EPYC 7543 (Zen 3) w/ Linux 4 on ESXi 7.0			
Intel Xeon D-1700/2700 w/Linux 4 on ESXi 7.0	Intel Xeon D-1746TER (Ice lake) w/ Linux 4 on ESXi 7.0		UCSC-E1100D-M6	
	Intel Xeon D-2796TE (Ice lake) w/ Linux 4 on ESXi 7.0			

Table 32: Algorithm Certificate Numbers

Algorithm	SFR	CiscoSSL FOM Cryptographic Implementation 7.3a (FTD[5])	CiscoSSL FOM Cryptographic Implementation 7.3a (FXOS[6])	CiscoSSL FOM Cryptographic Implementation 7.3a (FMC)	CiscoSSL FOM – Virtual 7.3a (FMCv)	Nitrox III series die	Nitrox-V GC
AES CBC 128/192/256 GCM 128/192/256	FCS_COP.1/DataEncryption	A4446	A4446	A4446	A4595	2034 2035	C1026
RSA 2048/3072 bits Signature Gen & Verify Key Gen & Verify	FCS_COP.1/SigGen FCS_CKM.1 FCS_CKM.1/IKE[VPN]	A4446	A4446	A4446	A4595	n/a	n/a
RSA key establishment	FCS_CKM.2	Tested with known good implementation	Tested with known good implementation	Tested with known good implementation	Tested with known good implementation	Tested with known good implementation	Tested with known good implementation
ECDSA curves P-256, P-384 and P-521 Key Sizes – 256, 384 and 521 bits	FCS_COP.1/SigGen FCS_CKM.1 FCS_CKM.1/IKE[VPN]	A4446	A4446	A4446	A4595	n/a	n/a

[5] Each Firepower appliance includes two instances of FOM 7.3a; one for FTD and one for FX-OS (underlying OS for FTD). Only one instance of the FOM was tested because they both run on the same processor.

[6] FXOS is running FOM 7.3a on the Supervisor Blade or MIO.

Algorithm	SFR	CiscoSSL FOM Cryptographic Implementation 7.3a (FTD[5])	CiscoSSL FOM Cryptographic Implementation 7.3a (FXOS[6])	CiscoSSL FOM Cryptographic Implementation 7.3a (FMC)	CiscoSSL FOM – Virtual 7.3a (FMCv)	Nitrox III series die	Nitrox-V GC
Signature Gen & Verify Key Gen and Verify							
FFC Scheme using key sizes of 2048-bit or greater DSA KeyPairGen	FCS_CKM.1	A4446	A4446	A4446	A4595	n/a	n/a
Hashing SHA-1, SHA-256, SHA-384, SHA-512	FCS_COP.1/Hash	A4446	A4446	A4446	A4595	1780	C1026
Keyed Hash HMAC-SHA-1, HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	FCS_COP.1/KeyedHash	A4446	A4446	A4446	A4595	1233	C1026
DRBG (Key Size – 256) HMAC_DRBG	FCS_RBG_EXT.1	A4446	A4446	A4446	A4595	n/a	n/a
KAS ECC	FCS_CKM.2	A4446	A4446	A4446	A4595	n/a	n/a

Algorithm	SFR	CiscoSSL FOM Cryptographic Implementation 7.3a (FTD[5])	CiscoSSL FOM Cryptographic Implementation 7.3a (FXOS[6])	CiscoSSL FOM Cryptographic Implementation 7.3a (FMC)	CiscoSSL FOM – Virtual 7.3a (FMCv)	Nitrox III series die	Nitrox-V GC
KAS FCC	FCS_CKM.2	A4446	A4446	A4446	A4595	n/a	n/a
FFC Schemes using 'safe-prime' groups	FCS_CKM.1 FCS_CKM.2	Tested with known good implementation	Tested with known good implementation	Tested with known good implementation	Tested with known good implementation	n/a	n/a

8 ANNEX A: REFERENCES

The following documentation was used to prepare this ST:

Table 33: References

Identifier	Description
[CC_PART1]	Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated April 2017, Version 3.1 Revision 5, CCMB-2017-04-001
[CC_PART2]	Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated April 2017, Version 3.1 Revision 5, CCMB-2017-04-002
[CC_PART3]	Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated April 2017, Version 3.1 Revision 5, CCMB-2017-04-003
[CEM]	Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated April 2017, Version 3.1 Revision 5, CCMB-2017-04-004
[800-38A]	NIST Special Publication 800-38A Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001
[800-56A]	NIST Special Publication 800-56A, March, 2007 Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised)
[800-56B]	NIST Special Publication 800-56B Recommendation for Pair-Wise, August 2009 Key Establishment Schemes Using Integer Factorization Cryptography
[FIPS 140-2]	FIPS PUB 140-2 Federal Information Processing Standards Publication Security Requirements for Cryptographic Modules May 25, 2001
[FIPS PUB 186-4]	FIPS PUB 186-3 Federal Information Processing Standards Publication Digital Signature Standard (DSS) July 2013
[FIPS PUB 186-5]	Federal Information Processing Standards Publication 186-5 Digital Signature Standard (DSS) February 3, 2023
[FIPS PUB 198-1]	Federal Information Processing Standards Publication The Keyed-Hash Message Authentication Code (HMAC) July 2008
[800-90]	NIST Special Publication 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators January 2012
[FIPS PUB 180-4]	FIPS PUB 180-4 Federal Information Processing Standards Publication Secure Hash Standard (SHS) March 2012

9 ANNEX B: SFR TOE COMPONENTS MAPPING

The following mapping was provided to show which SFR are supported by which TOE component:

Table 34: SFR Mapping

Requirement	Description	Distributed TOE SFR Allocation	Distributed TOE Audit Generation
Reproduced from NDcPP			
FAU_GEN.1	Audit Data Generation	All	All (startup/shutdown, and admin actions)
FAU_GEN.2	User Identity Association	All	N/A
FAU_GEN_EXT.1	Security Audit Generation	All	N/A
FAU_STG_EXT.1	Protected Audit Event Storage	All	N/A
FAU_STG_EXT.4	Protected Local Audit Event Storage for Distributed TOEs	All	N/A
FAU_STG_EXT.5	Protected Remote Audit Event Storage for Distributed TOEs	All	N/A
FCO_CPC_EXT.1	Communication Partner Control	FMC, FTD	FMC, FTD
FCS_CKM.1	Cryptographic Key Generation	All	N/A
FCS_CKM.2	Cryptographic Key Establishment	All	N/A
FCS_CKM.4	Cryptographic Key Destruction	All	N/A
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)	All	N/A
FCS_COP.1/SigGen	Cryptographic Operation (Signature Verification)	All	N/A
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)	All	N/A
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)	All	N/A
FCS_HTTPS_EXT.1	Protocol Feature Dependent	FMC and FXOS	FMC and FXOS
FCS_IPSEC_EXT.1(1)	IPsec Protocol – FXOS	FXOS	FXOS
FCS_NTP_EXT.1(1)	NTP Protocol - FXOS	FXOS	FXOS
FCS_NTP_EXT.1(2)	NTP Protocol - FMC	FMC	FMC
FCS_RBG_EXT.1	Random Bit Generation	All	N/A
FCS_SSHS_EXT.1(1)	SSH Server Protocol (FXOS)	FXOS	FXOS
FCS_SSHS_EXT.1(2)	SSH Server Protocol (FTD/FMC/FMCv)	FMC, FTD	FMC, FTD
FCS_TLSC_EXT.1	TLS Client	FMC, FTD	FMC, FTD
FCS_TLSC_EXT.2	TLS Client with authentication	FMC, FTD	FMC, FTD
FCS_TLSS_EXT.1	TLS Server	All	All
FCS_TLSS_EXT.2	TLS Server support for mutual authentication	FMC	FMC

Requirement	Description	Distributed TOE SFR Allocation	Distributed TOE Audit Generation
FIA_AFL.1	Authentication Failure Management	All	All
FIA_PMG_EXT.1	Password Management	All	N/A
FIA_UIA_EXT.1	User Identification and Authentication	All	All
FIA_UAU_EXT.2	Password-based Authentication Mechanism	All	All
FIA_UAU.7	Protected Authentication Feedback	All	N/A
FIA_X509_EXT.1/ITT FIA_X509_EXT.1/Rev	X.509 Certification Validation	All	All
FIA_X509_EXT.2(1) FIA_X509_EXT.2(2)	X.509 Certificate Authentication	All	N/A
FIA_X509_EXT.3	Certificate Requests	All	N/A
FMT_MOF.1/ManualUpdate	Trusted Update - Management of Security Functions behaviour	All	All
FMT_MOF.1/Services	Management of Security Functions behaviour	All	All
FMT_MTD.1/CoreData	Management of TSF Data	All	N/A
FMT_MTD.1/CryptoKeys	Management of TSF Data	All	N/A
FMT_SMF.1	Specification of Management Functions	FMC (*full*) FXOS (*subset*) FTD (*subset*) *(See TSS for details.)*	All
FMT_SMR.2	Restrictions on Security Roles	All	N/A
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys	All	N/A
FPT_APW_EXT.1	Protection of Administrator Passwords	All	N/A
FPT_TST_EXT.1	Testing (Extended)	All	N/A
FPT_ITT.1	Basic internal TSF data transfer protection	FMC, FTD	FMC, FTD
FPT_STM_EXT.1	Reliable Time Stamps	All	All
FPT_TUD_EXT.1	Trusted Update	All	All
FTA_SSL_EXT.1	TSF-Initiated Session Locking	All	All
FTA_SSL.3	TSF-initiated Termination	All	All
FTA_SSL.4	User-Initiated Termination	All	All
FTA_TAB.1	Default TOE Access Banner	All	N/A
FTP_ITC.1	Inter-TSF Trusted Channel	All	All
FTP_TRP.1/Admin	Trusted Path	All	All
Reproduced from mod_ips_v1.0			
FAU_GEN.1/IPS[IPS]	Audit Data Generation (IPS)	FMC, FTD	FMC, FTD
FAU_SAR.1[IPS]	Audit Review (IPS Data)	FMC	N/A
FAU_SAR.2[IPS]	Restricted Audit Review (IPS Data)	FMC	N/A
FAU_SAR.3[IPS]	Selectable Audit Review (IPS Data)	FMC	N/A

Requirement	Description	Distributed TOE SFR Allocation	Distributed TOE Audit Generation
FAU_STG.1[IPS]	Protected Audit Trail Storage (IPS Data)	FMC	N/A
FMT_SMF.1/IPS[IPS]	Specification of Management Functions (IPS)	FMC, FTD	FMC, FTD
IPS_ABD_EXT.1[IPS]	Anomaly-Based IPS Functionality	FTD	FTD
IPS_IPB_EXT.1[IPS]	IP Blocking	FTD	FTD
IPS_NTA_EXT.1[IPS]	Network Traffic Analysis	FTD, FMC	FMC
IPS SBD_EXT.1[IPS}	Signature-Based IPS Functionality	FTD	FTD
Reproduced from mod_cpp_fw_v1.4e			
FDP_RIP.2[FW]	Full Residual Information Protection	FTD	N/A
FFW_RUL_EXT.1[FW]	Stateful Traffic Filtering	FTD	FTD
FFW_RUL_EXT.2[FW]	Stateful Filtering of Dynamic Protocols	FTD	FTD
FMT_SMF.1/FFW[FW]	Specification of Management Functions	FMC, FTD	FMC, FTD
Reproduced from mod_vpngw_v1.3			
FAU_GEN.1/VPN[VPN]	Audit Data Generation (VPN Gateway)	FTD	N/A
FCS_CKM.1/IKE[VPN]	Cryptographic Key Generation (for IKE Peer Authentication)	FTD	N/A
FCS_IPSEC_EXT.1(2)[VPN]	IPsec Protocol - FTD	FTD	FTD
FMT_SMF.1/VPN[VPN]	Specification of Management Functions (VPN Gateway)	FTD, FMC	FTD, FMC
FPF_RUL_EXT.1[VPN]	Packet Filtering	FTD	FTD
FPT_FLS.1/SelfTest[VPN]	Fail Secure	FTD	FTD
FPT_TST_EXT.3[VPN]	Extended: TSF Testing	FTD	FTD
FTA_SSL.3[VPN]	TSF-initiated Termination	FTD	FTD
FTA_TSE.1[VPN]	TOE Session Establishment	FTD	FTD
FTA_VCM_EXT.1[VPN]	VPN Client Management	FTD	FTD
FTP_ITC.1/VPN[VPN]	Inter-TSF Trusted Channel	FTD	FTD