Cisco Adaptive Security Appliances (ASA) 9.16 on Firepower 1000 and 2100 Series

Security Target

ST Version 0.5

May 18, 2022

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
AAA	Administration, Authorization, and Accounting
ACL	Access Control List
AES	Advanced Encryption Standard
ASA	Adaptive Security Appliance
ASDM	Adaptive Security Device Manager
CC	Common Criteria
CEM	Common Evaluation Methodology
CM	Configuration Management
DHCP	Dynamic Host Configuration Protocol
EAL	Evaluation Assurance Level
ESP	Encapsulating Security Payload
FOM	FIPS Object Module
Gbps	Gigabits per second
GE	Gigabit Ethernet port
HTTPS	Hyper-Text Transport Protocol Secure
ICMP	Internet Control Message Protocol
IKE	Internet Key Exchange
IPsec	Internet Protocol Security
ISR	Integrated Services Router
IT	Information Technology
NDcPP	Network Device Collaborative Protection Profile
OS	Operating System
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)
ST	Security Target
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

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 Cisco Adaptive Security Appliance (ASA) on Firepower 2100 and 1000 Series. 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 Adaptive Security Appliance (ASA) 9.16 on Firepower 1000 and 2100 Series Security Target
ST Version	0.5
Publication Date	May 18, 2022
Vendor and ST Author	Cisco Systems, Inc.
TOE Reference	Cisco Adaptive Security Appliance (ASA) 9.16 on Firepower 1000 and 2100 Series
TOE Hardware Models	• Firepower 1000 Series (1010, 1120, 1140 and 1150) • Firepower 2100 Series (2110, 2120, 2130 and 2140)
TOE Software Version	ASA 9.16
Keywords	Firewall, VPN Gateway, Router

1.2 TOE Overview

The Cisco Firepower 1000 and 2100 Series Appliances are purpose-built, firewall platforms with VPN capabilities. The TOE includes the hardware models as defined in Table 2 of section 1.1.

1.2.1 TOE Product Type

The TOE is a purpose-built network device with firewall and VPN gateway capabilities. The firewall and VPN Gateway capabilities are as defined in the PP-Configuration for Network Devices, Stateful Traffic Filter Firewalls, and Virtual Private Network (VPN) Gateways (CFG_NDcPP-FW-VPNGW_V1.1). The TOE consists of hardware and software that provide connectivity and security services onto a single, secure device.

For firewall services, the ASA on and FP1000 and FP2100 all provide 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. Other uses refer to the use of IPsec connections to tunnel traffic that originates from or terminates at the TOE itself, such as for transmissions from the TOE to remote audit/syslog servers, or AAA servers, or for an additional layer of security for remote administration connections to the TOE, such as SSH or TLS connections tunneled in IPsec.

The TOE can operate in a number of modes: as a single standalone device, or in high-availability (HA) failover-pairs; with a single-context, or with multiple-contexts within each single/pair; as a transparent firewall when deployed in single-context, or with one or more contexts connected to two or many IP subnets when configured in router mode.

For management purposes, the ASDM is included. ASDM allows the TOE to be managed from a graphical user interface.

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

1.2.2 Supported non-TOE Hardware/ Software/ Firmware

The TOE supports (in some cases optionally) the hardware, software, and firmware listed in Table 3 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.
Local Console	Yes	The Console that is directly connected to the TOE via the Serial Console Port and is used by the TOE administrator to support TOE administration.
ASDM Management Platform	Yes	The ASDM operates from any of the following operating systems: • Microsoft Windows 7, 8, 10, Server 2008, Server 2012, and Server 2012 R2 • Apple OS X 10.4 and later Note that that ASDM software is downloaded from the TOE, and runs on the management platform but is not installed to the management platform, nor is ASDM installed to the TOE. ASDM runs on the management platform is used to connect to the TOE over TLS. The only software installed on the management platform is a Cisco ASDM Launcher.
Web browser	Yes	The following web browsers are supported for access to the ASDM; • Internet Explorer • Firefox • Safari • Chrome Note: Using the latest supported web browser version is recommended.
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.
AAA Server	No	This includes any IT environment AAA server that provides single-use authentication mechanisms. The TOE correctly leverages the services provided by this AAA server to provide single-use authentication to administrators. Connections to remote AAA servers must be tunneled in IPsec.
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.

Component	Required	Usage/Purpose Description for TOE performance
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 (e.g., Cisco AnyConnect, Cisco VPN client) that supports IPsec tunneling. Both VPN clients and VPN gateways can be considered to be remote tunnel endpoints.

1.3 TOE DESCRIPTION

This section provides an overview and description of the TOE. The TOE is comprised of both software and hardware. The model is comprised of the following: FP1010, FP1120, FP1140, FP1150, FP2110, FP2120, FP2130, and FP2140. The software is comprised of the Adaptive Security Appliance software image Release 9.16. The ASA software provides all firewall and VPN functionality.

The Cisco Adaptive Security Appliances that comprise the TOE have common hardware characteristics. One the other hand, the 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 ASAs in terms of hardware.

Figure 1: FP2100 Series Hardware (2110, 2120, 2130, 2140)

The ASA FP2100 have the following distinct characteristics:

Table 4: FP2100 Series Hardware

Model	FP2110	FP2120	FP2130	FP2140
Number of Processors	1	1	1	1
Processor	Intel Xeon D-1526 (Broadwell)	Intel Xeon D-1528 (Broadwell)	Intel Xeon D-1548 (Broadwell)	Intel Xeon D-1577 (Broadwell)
	OCTEON III CN7230 MIPS64 (cryptographic accelerator for IPsec implementation)	OCTEON III CN7340 MIPS64 (cryptographic accelerator for IPsec implementation)	OCTEON III CN7350 MIPS64 (cryptographic accelerator for IPsec implementation)	OCTEON III CN7360 MIPS64 (cryptographic accelerator for IPsec implementation)

Storage	100 GB	100 GB	200 GB	200 GB
Integrated I/O	12 x 10M/100M/1GBASE-T Ethernet interfaces (RJ-45), 4 x 1 Gigabit (SFP) Ethernet interfaces	12 x 10M/100M/1GBASE-T Ethernet interfaces (RJ-45), 4 x 1 Gigabit (SFP) Ethernet interfaces	12 x 10M/100M/1GBASE-T Ethernet interfaces (RJ-45), 4 x 10 Gigabit (SFP+) Ethernet interfaces	
IPSec VPN Throughput	750 Mbps	1000 Mbps	1500 Mbps	3000 Mbps
Maximum concurrent sessions	1 million	1.2 million	2 million	3.0 million
Maximum VPN Peers	1500	3500	7500	10000

Figure 2: FP1000 Series Hardware (1010)

Figure 3: FP1000 Series Hardware (1120, 1140)

Figure 4: FP1000 Series Hardware (1150)

The FP1000 models have the following distinct characteristics:

Table 5: FP1000 Series Hardware

Model	FP1010	FP1120	FP1140	FP1150
Number of Processors	1	1	1	1
Processor(s)	Intel Atom C3558 (Goldmont)	Intel Atom C3858 (Goldmont)	Intel Atom C3958 (Goldmont)	Intel Atom C3958 (Goldmont)
Storage	1 x 200 GB	1 x 200 GB	1 x 200 GB	1 x 200 GB
Integrated I/O	8 x RJ45, 4x SFP	8 x RJ45, 4x SFP	8 x RJ45, 4x SFP	8 x RJ-45, 2 x 1Gbps SFP, 2 x 1/10Gbps SFP+
IPsec VPN Throughput	300 Mbps	1 Gbps	1.2 Gbps	1.7 Gbps
Maximum concurrent sessions	100K	200K	400K	600K
Maximum VPN Peers	75	150	400	800

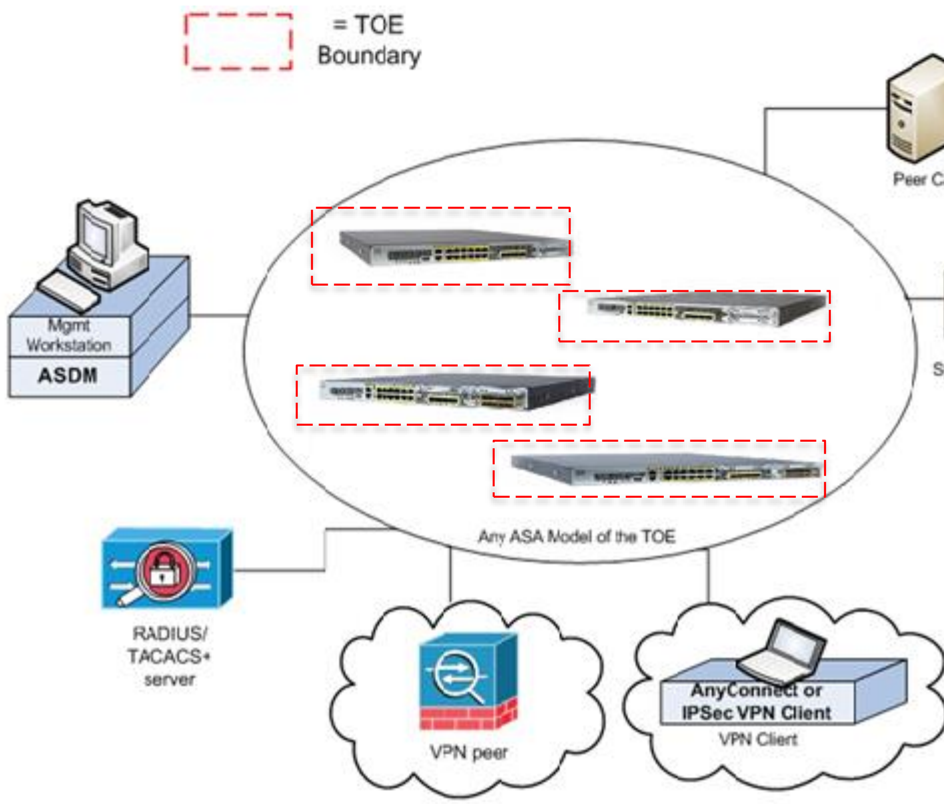
1.4 TOE Evaluated Configuration

The TOE consists of one or more physical devices as specified in section 1.5 below. Each physical device includes the Cisco ASA software. Each instantiation of the TOE has two or more network interfaces and can filter IP traffic to and through those interfaces.

If the TOE is to be remotely administered, the management station must connect using SSHv2. When ASDM 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. The TOE is able to filter connections to/from these external entities using its IP traffic filtering and can encrypt traffic where necessary using TLS and/or IPsec.

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

Figure 5: Example TOE Deployment



The previous figure includes the following:

- Several examples of TOE Models
- VPN Peer (Operational Environment) or another instance of the TOE
- VPN Peer (Operational Environment) with an IPsec VPN Client.
- Management Workstation (Operational Environment) with ASDM and SSH Client
- Remote Authentication Server (Operational Environment)
- Peer CA (Operational Environment)
- Syslog server (Operational Environment)

1.5 Physical Scope of the TOE

The TOE is a hardware and software solution as listed in Table 6:

Table 6: Hardware Models and Specifications

TOE Configuration	Hardware Configurations	Software Version
FP1010 **FP1120** **FP1140** **FP1150**	The Cisco ASA Adaptive Security Appliance on FP1000 provides high-performance firewall and VPN services	ASA release 9.16

FP2110 **FP2120** **FP2130** **FP2140**	The Cisco ASA Adaptive Security Appliance on FP2100 provides high-performance firewall and VPN services	ASA release 9.16

1.6 Logical Scope of the TOE

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

1. Security Audit
2. Cryptographic Support
3. Full Residual Information Protection
4. Identification and Authentication
5. Security Management
6. Protection of the TSF
7. TOE Access
8. Trusted Path/Channels
9. Filtering

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 newest audit record will overwrite the oldest audit record when the local storage space for audit data is full. Audit logs are backed up over an encrypted channel to an external audit server, if so configured.

1.6.2 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 entropy noise source.

1.6.3 Full Residual Information 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.

1.6.4 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. 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 or pre-shared key methods.

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 of 3-127 characters. The TOE also implements a lockout mechanism if the number of configured unsuccessful authentication attempts has been exceeded.

The TOE provides administrator authentication against a local user database. Password-based authentication can be performed on the serial console, SSHv2, and HTTPS interfaces. The TOE optionally supports use of AAA server including RADIUS and TACACS+ (part of the IT Environment) for authentication of administrative users attempting to connect to the TOE.

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

1.6.6 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. 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.

1.6.7 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.

1.6.8 Trusted path/Channels

The TOE supports establishing trusted paths between itself and remote administrators using SSHv2 for CLI access, and TLS/HTTPS for GUI/ASDM access apart from tunneling the ASDM and/or SSH connections in IPsec VPN tunnels. The TOE supports use of TLS and/or IPsec for connections with remote syslog servers. The TOE can use IPsec to encrypt connections with remote authentication servers (e.g. RADIUS, TACACS+). 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.

1.6.9 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 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 crypto map set.

1.7 Excluded Functionality

The following functionality is excluded from the evaluation.

Table 7: Excluded Functionality

Excluded Functionality	Exclusion Rationale
Telnet for management purposes	Telnet passes authentication credentials in clear text and is disabled by default.
Secure Policy Manager is excluded from the evaluated configuration	Use of Security Policy Manager 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.
ASA Cluster functionality is excluded from the evaluated configuration.	Use of ASA Clustering is beyond the scope of this Common Criteria evaluation.

These services will be disabled by configuration. The exclusion of this functionality does not affect compliance to the collaborative Protection Profile for Network Devices (cpp_nd_v2.2e), PP-Module for Stateful Traffic Filter Firewalls (mod_cpp_fw_v1.4e) and PP-Module for Virtual Private Network (VPN) Gateways (mod_vpngw_v1.1).

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 5.7.

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 8 below:

Table 8: Protection Profiles

Protection Profile	Version	Date
PP-Configuration for Network Devices, Stateful Traffic Filter Firewalls, and Virtual Private Network (VPN) Gateways (CFG_NDcPP-FW-VPNGW_V1.1)	1.1	1 July 2020
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 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_V1.1)	1.1	18 June 2020

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 9: Technical Decisions

TD #	TD Name	Protection Profiles	Applied to this TOE
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
TD0634	NIT Technical Decision for Clarification required for testing IPv6	CPP_ND_V2.2E	FCS_TLSC_EXT.1.2
TD0633	NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	CPP_ND_V2.2E	FCS_IPSEC_EXT.1
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

TD0597	VPN GW IPv6 Protocol Support	MOD_VPNGW_v1.1	FPF_RUL_EXT.1.6
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
TD0590	Mapping of operational environment objectives	MOD_VPNGW_v1.1	Section 4.3 of PP module
TD0581	NIT Technical Decision for Elliptic curve based key establishment and NIST SP 800-56A rev 3	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.1.4, Test 3
TD0555	NIT Technical Decision for RFC Reference incorrect in TLSS Test	CPP_ND_V2.2E	FCS_TLSS_EXT.1.4, Test 3
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
TD0549	Consistency of Security Problem Definition update for MOD_VPNGW_v1.0 and MOD_VPNGW_v1.1	MOD_VPNGW_v1.1	Assumption – A.CONNECTIONS
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
TD0538	NIT Technical Decision for Outdated link to allowed-with list	CPP_ND_V2.2E	Section 2 of PP
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*	*Not applied because this ST does not include FCS_NTP_EXT.1*

TD0527	Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	CPP_ND_V2.2E	FIA_X509_EXT.1/REV, FIA_X509_EXT.1/ITT

2.2.1 Protection Profile Additions or Modifications

The following requirements were modified:

- FAU_GEN.1 – Additional auditable events were added from mod_cpp_fw_v1.4e and mod_vpngw_v1.1
- FCS_COP.1/DataEncryption - This SFR has been modified from its definition in the NDcPP to support mod_vpngw_v1.1's IPsec requirements by mandating support for at least one of CBC or GCM modes and at least one of 128-bit or 256-bit key sizes at minimum.
- FCS_IPSEC_EXT.1 – This SFR has been modified from its definition in the NDcPP to support mod_vpngw_v1.1's requirements, since IPsec is used to implement the VPN functionality required by mod_vpngw_v1.1.
- FIA_X509_EXT.2 – This SFR has been modified since it is mandatory for any TOE that claims conformance to mod_vpngw_v1.1 because a conformant TOE will always have the ability to receive an X.509 certificate from an external entity as part of IPsec communications.
- FIA_X509_EXT.3 - This SFR has been modified since it is mandatory for any TOE that claims conformance to mod_vpngw_v1.1. A conformant TOE will always have the ability to receive an X.509 certificate from an external entity as part of IPsec communications.
- FMT_MTD.1/CryptoKeys – This SFR has been refined to refer specifically to keys and certificates used for VPN operation.
- FMT_SMF.1 – This SFR has been modified to conform to the MOD_VPNGW_V1.1 requirements.
- FPT_TST_EXT.1 – This SFR has been modified from its definition in the NDcPP by requiring noise source health tests to be performed regardless of what other testing is claimed.
- FPT_TUD_EXT.1 – This SFR has been modified from its definition in the NDcPP because the MOD_VPNGW_V1.1 requires the digital signature method to be selected at a minimum.
- FDP_RIP.2[FW], FFW_RUL_EXT.1[FW], FFW_RUL_EXT.2[FW] and FMT_SMF.1/FFW[FW] – These SFRs were added to conform to MOD_CPP_FW_1.4E.
- FCS_CKM.1/IKE[VPN], FIA_PSK_EXT.1[VPN], FMT_SMF.1/VPN[VPN], FPF_RUL_EXT.1[VPN], FPT_FLS.1/SelfTest[VPN], FPT_TST_EXT.3[VPN], FTA_SSL.3/VPN[VPN], FTA_TSE.1[VPN], FTA_VCM_EXT.1[VPN] and FTP_ITC.1/VPN[VPN] - These SFRs were added to conform to MOD_VPNGW_V1.1

2.3 Protection Profile Conformance Claim Rationale

2.3.1 TOE Appropriateness

The TOE provides all of 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 for Stateful Traffic Filter Firewalls (mod_cpp_fw_v1.4e); and
- PP-Module for Virtual Private Network (VPN) Gateways (mod_vpngw_v1.1)

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 cpp_nd_v2.2e, mod_cpp_fw_v1.4e and mod_vpngw_v1.1 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 Collaborative Protection Profile 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_cpp_fw_v1.4e and mod_vpngw_v1.1 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 cpp_nd_v2.2e, section 7 of mod_cpp_fw_v1.4e and section 5.4 of mod_vpngw_v1.1.

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 10: 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.

Assumption	Assumption Definition
A.NO_THRU_TRAFFIC_PROTECTION	A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the NDcPP. It is assumed that this protection will be covered by cPPs and PP-modules for particular types of network devices (e.g., firewall).
A.TRUSTED_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.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 the mod_cpp_fw_v1.4e	
Same as base cPP (NDcPP v2.2e)	
Reproduced from the mod_vpngw_v1.1	
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 11: Threats

Threat	Threat Definition
Reproduced from the 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.

Threat	Threat Definition
T.WEAK_CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_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.

Threat	Threat Definition
T.PASSWORD_CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic and may allow them to take advantage of any trust relationships with other network devices.
T.SECURITY_FUNCTIONALITY_ FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.
Reproduced from the mod_cpp_fw_v1.4e	
T.NETWORK_DISCLOSURE	An attacker may attempt to “map” a subnet to determine the machines that reside on the network, and obtaining the IP addresses of machines, as well as the services (ports) those machines are offering. This information could be used to mount attacks to those machines via the services that are exported.
T.NETWORK_ACCESS	With knowledge of the services that are exported by machines on a subnet, an attacker may attempt to exploit those services by mounting attacks against those services.
T.NETWORK_MISUSE	An attacker may attempt to use services that are exported by machines in a way that is unintended by a site’s security policies. For example, an attacker might be able to use a service to “anonymize” the attacker’s machine as they mount attacks against others.
T.MALICIOUS_TRAFFIC	An attacker may attempt to send malformed packets to a machine in hopes of causing the network stack or services listening on UDP/TCP ports of the target machine to crash.
Reproduced from the mod_vpngw_v1.1	
T.DATA_INTEGRITY	Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to modify the data without authorization. If known malicious external devices are able to communicate with devices on the protected network or if devices on the protected network can establish communications with those external devices then the data contained within the communications may be susceptible to a loss of integrity.

Threat	Threat Definition
T. NETWORK_ACCESS	Devices located outside the protected network may seek to exercise services located on the protected network that are intended to only be accessed from inside the protected network or only accessed by entities using an authenticated path into the protected network. Devices located outside the protected network may, likewise, offer services that are inappropriate for access from within the protected network. From an ingress perspective, VPN gateways can be configured so that only those network servers intended for external consumption by entities operating on a trusted network (e.g., machines operating on a network where the peer VPN gateways are supporting the connection) are accessible and only via the intended ports. This serves to mitigate the potential for network entities outside a protected network to access network servers or services intended only for consumption or access inside a protected network. From an egress perspective, VPN gateways can be configured so that only specific external services (e.g., based on destination port) can be accessed from within a protected network, or moreover are accessed via an encrypted channel. For example, access to external mail services can be blocked to enforce corporate policies against accessing uncontrolled e-mail servers, or, that access to the mail server must be done over an encrypted link.

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

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

3.3 Organizational Security Policies

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

Table 12: Organizational Security Policies

Policy Name	Policy Definition
Reproduced from the 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.

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 13: Security Objectives for the TOE

TOE Objective	TOE Security Objective Definition
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 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_v1.1	
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.

TOE Objective	TOE Security Objective Definition
O.AUTHENTICATION	To further address the issues associated with unauthorized disclosure of information, a compliant TOE's authentication ability (IPSec) will allow a VPN peer to establish VPN connectivity with another VPN peer. VPN endpoints authenticate each other to ensure they are communicating with an authorized external IT entity.
O.CRYPTOGRAPHIC_FUNCTIONS	To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption of services, and network-based reconnaissance, compliant TOE's will implement a cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.
O.FAIL_SECURE	There may be instances where the TOE's hardware malfunctions or the integrity of the TOE's software is compromised, the latter being due to malicious or non-malicious intent. To address the concern of the TOE operating outside of its hardware or software specification, the TOE will shut down upon discovery of a problem reported via the self-test mechanism and provide signature-based validation of updates to the TSF.
O.PORT_FILTERING	To further address the issues associated with unauthorized disclosure of information, etc., a compliant TOE's port filtering capability will restrict the flow of network traffic between protected networks and other attached networks based on the originating (source) and/or receiving (destination) port (or service) identified in the network traffic as well as on established connection information.
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).

TOE Objective	TOE Security Objective Definition
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 14: Security Objectives for the Environment

Environment Security Objective	IT Environment Security Objective Definition
Reproduced from the 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.

Environment Security Objective	IT Environment Security Objective Definition
OE.UPDATES	The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_ SECURE	The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.
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_vpngw_v1.1	
OE.CONNECTIONS	The TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks.

5 SECURITY REQUIREMENTS

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;
- Refinement made by ST author: Indicated with <u>**bold and underlined**</u> text;
- Refinement made in the PP: the refinement text is indicated with bold text and ~~strikethroughs~~;
- Selection: Indicated with <u>underlined</u> 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_cpp_fw_v1.4e and mod_vpngw_v1.1 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 EP themselves. In addition, SFRs copied verbatim from mod_cpp_fw_v1.4e will have an extension [FW], SFRs copied from mod_vpngw_v1.1 will have extension [VPN] to distinguish them from the cpp_nd_v2.2e. These SFRs that have an extension of [FW] or [VPN] do not exist in cpp_nd_v2.2e. Changes have been made to the base cPP SFRs as necessary to support the firewall and VPN functionality based on mod_cpp_fw_v1.4e and mod_vpngw_v1.1.

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 15: Security Functional Requirements

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

Class Name	Component Identification	Component Name
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	IPsec Protocol
	FCS_RBG_EXT.1	Random Bit Generation
	FCS_SSHS_EXT.1	SSH Server Protocol
	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
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/Rev	X.509 Certificate Validation
	FIA_X509_EXT.2	X.509 Certificate Authentication
	FIA_X509_EXT.3	X.509 Certificate Requests
FMT: Security Management	FMT_MOF.1/ManualUpdate	Management of security functions behaviour

Class Name	Component Identification	Component Name
	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 (Extended)
	FPT_TUD_EXT.1	Trusted Update
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_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
Reproduced from mod_vpngw_v1.1		
FCS: Cryptographic Support	FCS_CKM.1/IKE[VPN]	Cryptographic Key Generation (for IKE Peer Authentication)

Class Name	Component Identification	Component Name
FIA: Identification and Authentication	FIA_PSK_EXT.1[VPN]	Pre-Shared Key Composition
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 cpp_nd_v2.2e

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 not specified level of audit; and

c) All administrative actions comprising:

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

d) *Specifically defined auditable events listed in Table 16**Error! Reference source not found.**.*

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* ***Error! Reference source not found.****Table 16.*

Table 16: Auditable Events

SFR	Auditable Event	Additional Audit Record Contents
Reproduced from the NDcPP		
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1	Failure to establish an HTTPS session.	Reason for failure
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA. Session Establishment with peer	Reason for failure Entire packet contents of packets transmitted/received during session establishment
FCS_RBG_EXT.1	None.	None.
FCS_SSHS_EXT.1	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 an TLS Session	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/Rev	• Unsuccessful attempt to validate a certificate	• Reason for failure of certificate validation

SFR	Auditable Event	Additional Audit Record Contents
	• Any addition, replacement or removal of trust anchors in the TOE's trust store	• Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/ ManualUpdate	Any attempt to initiate a manual update	None.
FMT_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)	No additional information.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1	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.

SFR	Auditable Event	Additional Audit Record Contents
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 mod_vpngw_v1.1		
FCS_CKM.1/IKE[VPN]	None.	None.
FIA_PSK_EXT.1[VPN]	None.	None.
FMT_SMF.1/VPN[VPN]	None.	None.
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]	None.	None.
FPT_TST_EXT.3[VPN]	None.	None.
FTA_SSL.3/VPN[VPN]	None.	None.
FTA_TSE.1[VPN]	None.	None.
FTA_VCM_EXT.1[VPN]	None.	None.
FTP_ITC.1/VPN[VPN]	None.	None.

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_STG_EXT.1 Protected Audit Event Storage

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

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

- *The TOE shall consist of a single standalone component that stores audit data locally,*

]

FAU_STG_EXT.1.3 The TSF shall [*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.

5.3.2 Cryptographic Support (FCS)

5.3.2.1 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 cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1*
- *FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and* [*RFC 3526*]

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

5.3.2.2 FCS_CKM.2 Cryptographic Key Establishment

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

- *RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 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"*
- *Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 2, "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*].~~

5.3.2.3 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*.

5.3.2.4 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 [*no other cryptographic key sizes*]** that meet the following: AES as specified in ISO 18033-3, ***[CBC as specified in ISO 10116, GCM as specified in ISO 19772]*** **and [*no other standards*]**.

5.3.2.5 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 RSASSAPKCS2v1_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*

].

Application Note

IKE/IPsec supports both ECDSA and RSA digital signature. SSH and trusted update only support RSA digital signature.

5.3.2.6 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*.

5.3.2.7 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"*.

5.3.2.8 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.

5.3.2.9 FCS_IPSEC_EXT.1 IPsec Protocol

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

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

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

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

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [

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

].

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

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [

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

  [

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

  ]

].

Application Note

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.

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [

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

  [

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

  ]

].

Application Note

The valid range in kilobytes is 10 to 2,147,483,647 (10KB to 2TB).

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [*512*] bits.

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

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

] .

FCS_IPSEC_EXT.1.11 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 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 The TSF shall ensure that all IKE protocols perform peer authentication using [*RSA, ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [*Pre-shared Keys*].

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

5.3.2.10 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 [*Hash_DRBG (any), CTR_DRBG (AES)*].

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

Application Note

The TOE contains two separate entropy sources. Both entropy sources are described in detail in the proprietary entropy design documents.

5.3.2.11 FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFC(s) 4251, 4252, 4253, 4254, [6668].

FCS_SSHS_EXT.1.2 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 The TSF shall ensure that, as described in RFC 4253, packets greater than [*65,535 bytes*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [*aes128cbc, aes256-cbc*].

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

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

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

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

5.3.2.12 FCS_TLSC_EXT.1 TLS Client Protocol Without Mutual Authentication

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

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA *as defined in RFC 3268*
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA *as defined in RFC 3268*
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 *as defined in RFC 5246* ***(TLSv1.2-only)***
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 *as defined in RFC 5246* ***(TLSv1.2-only)***
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 *as defined in RFC 5289* ***(TLSv1.2-only)***
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 *as defined in RFC 5289* ***(TLSv1.2-only)***].

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*].

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.

5.3.2.13 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.

5.3.2.14 FCS_TLSS_EXT.1 TLS Server Protocol

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

- *TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
- *TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246* ***(TLSv1.2 only)***
- *TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246* ***(TLSv1.2 only)***
- *TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289* ***(TLSv1.2 only)***
- *TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289* ***(TLSv1.2 only)***].

and no other ciphersuites..

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

FCS_TLSS_EXT.1.3 The TSF shall perform key establishment for TLS using [*Diffie-Hellman parameters with size [2048 bits, 3072 bits]; ECDHE curves [secp256r1, secp384r1, secp521r1] and no other curves*].

FCS_TLSS_EXT.1.4 The TSF shall support [*session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2)*].

5.3.3 Identification and authentication (FIA)

5.3.3.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1 to 16*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password.*

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

5.3.3.2 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 [*3*] and [*127*] characters.

5.3.3.3 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.

5.3.3.4 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+]*] authentication mechanism to perform local administrative user authentication.

5.3.3.5 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**.

5.3.3.6 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, 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.

5.3.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication

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

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

5.3.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

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

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

5.3.4 Security management (FMT)

5.3.4.1 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*.

5.3.4.2 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*.

5.3.4.3 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*].

5.3.4.4 FMT_SMF.1 Specification of Management Functions

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

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

- *Ability to 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 manage the trusted public keys database;*

]

5.3.4.5 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.

5.3.5 Protection of the TSF (FPT)

5.3.5.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)

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

5.3.5.2 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.

5.3.5.3 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*].

5.3.5.1 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*].

5.3.5.2 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 TOE Access (FTA)

5.3.6.1 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.

5.3.6.2 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.*

5.3.6.3 FTA_SSL.4 User-initiated Termination

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

5.3.6.4 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.

5.3.7 Trusted Path/Channels (FTP)

5.3.7.1 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, [*authentication server, 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 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 or TLS;*
- *Authentication server: authentication of TOE administrators using AAA servers including RADIUS and TACACS+ over IPsec;*
- *Remote VPN peer using IPsec;*].

5.3.7.2 FTP_TRP.1/Admin Trusted Path

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

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

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

5.4 SFRs from mod_cpp_fw_v1.4e

5.4.1 User Data Protection (FDP)

5.4.1.1 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.

5.4.2 Stateful Traffic Filtering (FFW)

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

- [*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. *IPv4 packet with destination IP address equal to 0.0.0.0*
            - b. *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*

ii. *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.*

iii. *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.*

iv. *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].*

5.4.2.2 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*].

5.4.3 Security Management (FMT)

5.4.3.1 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*

5.5 SFRs from mod_vpngw_v1.1

5.5.1 Cryptographic Support (FCS)

5.5.1.1 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 186-4, "Digital Signature Standard (DSS)"; Appendix B.3 for RSA schemes;***
- ***FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4 for ECDSA schemes and implementing "NIST curves" P-256, P-384 and [P-521]***

]

and

[*no other key generation algorithms*]

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

5.5.2 Identification and authentication (FIA)

5.5.2.1 FIA_PSK_EXT.1[VPN] Pre-Shared Key Composition

FIA_PSK_EXT.1.1[VPN] The TSF shall be able to use pre-shared keys for IPsec and [*no other protocols*].

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

- Are 22 characters and [*[up to 128 characters]*];
- composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")").

FIA_PSK_EXT.1.3[VPN] The TSF shall condition the text-based pre-shared keys by using [*SHA-1, SHA-256, SHA-512, [SHA-384*].

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

5.5.3 Security Management (FMT)

5.5.3.1 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;*
  [
- *No other capabilities*
  ]]

5.5.4 Packet Filtering (FPF)

5.5.4.1 FPF_RUL_EXT.1[VPN] Rules for 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 2460)
  - Source address
  - Destination Address
  - Next Header (Protocol)
- TCP(RFC 793)
  - Source Port
  - Destination Port
- UDP(RFC 768)
  - Source Port
  - Destination Port

FPF_RUL_EXT.1.3[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.

5.5.5 Protection of the TSF (FPT)

5.5.5.1 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*].

5.5.5.2 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*].

5.5.6 TOE Access (FTA)

5.5.6.1 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*.]

5.5.6.2 FTA_TSE.1[VPN] TOE Session Establishment

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

5.5.6.3 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.

5.5.7 Trusted Path/Channels (FTP)

5.5.7.1 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/peers*].

5.6 TOE SFR Dependencies Rationale for SFRs Found in NDcPP

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

5.7 Security Assurance Requirements

5.7.1 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 17: 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

5.7.2 Security Assurance Requirements Rationale

This Security Target claims conformance to cpp_nd_v2.2e. 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_cpp_fw_v1.4e and mod_vpngw_v1.1, and 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.

5.8 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 18: 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.
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 19: 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 cpp_nd_v2.2e</td></tr>
<tr><td>FAU_GEN.1</td><td>
Shutdown and start-up of the audit functions are logged by events for reloading the TOE, and the events when the TOE comes back up. When audit is enabled, it is on whenever the TOE is on. Also, if logging is ever disabled, it is displayed in the ASDM Real-Time Log Viewer as a syslog disconnection and then a reconnection once it is re-established followed by an event that shows that the "logging enable" command was executed. TOE can enable or disable logging of all audit events and this is also logged. See the table within this cell for other required events and rationale.
The TOE generates events in the following format, with fields for date and time, type of event (the ASA-x-xxxxxx identifier code), subject identities, and outcome of the event:
e.g.:
Nov 21 2012 20:39:21: %ASA-3-713194: Group = 192.168.22.1, IP = 192.168.22.1, Sending IKE Delete With Reason message: Disconnected by Administrator.
The TOE 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, such as:
Nov 21 2012 20:39:21: %ASA-3-201011: Connection limit exceeded cnt/limit for dir packet from sip/sport to dip/dport on interface if_name.
Nov 21 2012 20:39:21: %ASA-3-202011: Connection limit exceeded econns/limit for dir packet from source_address/source_port to dest_address/dest_port on interface interface_name
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</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</td></tr>
</table>
</td></tr>
</table>

TOE SFRs	How the SFR is Satisfied	
	part of the authorized administrator role.	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.
	All use of the user identification mechanism.	Events will be generated for attempted identification/ authentication, and the username attempting to authenticate will be recorded in the event.
	Any use of the authentication mechanism.	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.
	The reaching of the threshold for unsuccessful authentication attempts and the subsequent restoration by the authorized administrator of the user's capability to authenticate.	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.
	All decisions on requests for information flow.	In order for events to be logged for information flow requests, the 'log' keyword must 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.
	Success and failure, and the type of cryptographic operation	Attempts for VPN connections are logged (whether successful or failed). Requests for encrypted session negotiation are logged (whether successful or failed). The identity of the user performing the cryptographic operation is included in the event. The unique key name is logged.
	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.

TOE SFRs	How the SFR is Satisfied	
	Changes to the time.	Changes to the time are logged with old and new time values.
	Use of the functions listed in this requirement 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, SSH, and IPsec.
	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.
	Indication of packets dropped due to too much network traffic	Logs are generated when traffic that exceeds the settings allowed on an interface is received.
	FTP Connection	Logs are generated for all FTP connections.
FAU_GEN.2	The TOE ensures each action performed by the administrator at the CLI, or via ASDM is logged with the administrator’s identity and as a result events are traceable to a specific user.	

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td>FAU_STG_EXT.1</td><td>The TOE can be configured to export syslog records to an administrator-specified, external syslog server in real-time. The TOE can be configured to encrypt the communications with an external syslog server using IPsec or TLS.

If using syslog through an IPsec tunnel, the TOE can be configured to block any new ‘permit’ actions that might occur. In other words, it can be configured to stop forwarding network traffic when it discovers it can no longer communicate with its configured syslog server(s).

The TOE will buffer syslog messages locally, but the local buffer will be cleared when the TOE is rebooted. The default size of the buffer is 4KB, and can be increased to 16KB. When the local buffer is full, the oldest message will be overwritten with new messages. Only authorized administrators can configure the local buffer size, reboot the TOE, and configure the external syslog server.</td></tr>
<tr><td>FCS_CKM.1, FCS_CKM.2</td><td>In the TOE cryptographic functions are used to establish TLS, HTTPS, IPsec, and SSH sessions, for IPsec traffic and authentication keys, and for IKE authentication and encryption keys.

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 (implements only required functions), “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, FFC schemes with key sizes 2048 and 3072 bits according to “Digital Signature Standard (DSS)”, Appendix B.1 and using Diffie-Hellman group 14 that meet Section 3 of RFC 3526.

Key establishment for asymmetric keys on all models of the TOE implements RSA-based, ECDSA-based and DH-based key establishment schemes as specified in Section 7.2 of RFC 3447, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1 and NIST SP 800-56A “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” respectively. In addition, the TOE also supports FFC schemes key establishment method that used groups listed in RFC 3526.

The TOE provides cryptographic signature services using RSA and ECDSA with key sizes (modulus) of 2048 and 3072 bits, and 256, 384, and 521 bits, respectively. For RSA, the key size is configurable down to 1024, but only 2048 key size or higher is permitted in the evaluated configuration. The key generation is also tested as part of the signature generation and verification functions.

The TOE supports key establishment including RSA-based, ECDSA-based and DH-based schemes. The KAS ECC and FFC + CVL algorithm testing information is provided in Section 7.3

<table>
<tr><th>Scheme</th><th>SFR</th><th>Services</th></tr>
<tr><td>RSA</td><td>FCS_IPSEC_EXT.1,
FCS_SSHS_EXT.1</td><td>SSH Remote Administration, syslog over IPsec, Connections with AAA servers over IPsec, IPsec VPN communication</td></tr>
<tr><td>ECC (P-256,
P-348,
P-521)</td><td>FCS_TLSC_EXT.1
FCS_TLSC_EXT.2
FCS_IPSEC_EXT.1</td><td>Syslog over TLS, Syslog over IPsec, Connections with AAA servers over IPsec, IPsec VPN communication</td></tr>
</table>
</td></tr>
</table>

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td></td><td><table>
<tr><td>ECC
(P-256,
P-348,
P-521)</td><td>FCS_TLSS_EXT.1</td><td>HTTPS Remote Administration</td></tr>
<tr><td>FFC</td><td>FCS_TLSC_EXT.1
FCS_TLSC_EXT.2</td><td>Syslog over TLS</td></tr>
<tr><td>FFC</td><td>FCS_TLSS_EXT.1</td><td>HTTPS Remote Administration</td></tr>
<tr><td>FFC using 'safe-prime' groups</td><td>FCS_SSHS_EXT.1
FCS_IPSEC_EXT.1</td><td>SSH Remote Administration, IKE communication</td></tr>
</table></td></tr>
<tr><td>FCS_CKM.4</td><td>The TOE meets all requirements specified in FIPS 140-2 for destruction of keys and Critical Security Parameters (CSPs). Additional key zeroization detail is provided in section 7.2. The relevant algorithms have been FIPS validated as indicated in section 7.3

An example of manually triggering zeroization is: existing RSA and ECDSA keys will be zeroized when new RSA and ECDSA keys are generated, and zeroization of RSA and ECDSA keys can be triggered manually through use of the commands:

asa(config)#crypto key zeroize rsa [label key-pair-label] [default] [noconfirm]

asa(config)#crypto key zeroize ec [label key-pair-label]</td></tr>
<tr><td>FCS_COP.1/ DataEncryption</td><td>Cryptographic functions are used to establish TLS, HTTPS, IPsec and SSH sessions, for IPsec traffic and authentication keys, IKE authentication and encryption keys and to establish SSH sessions and TLS sessions.

The TOE supports AES-CBC and AES-GCM, each with 128 or 256-bit (as described in ISO 18033-3 for AES, ISO 10116 for CBC mode and ISO 19772 for GCM mode). The TOE uses a FIPS-validated implementation of AES with 128 and 256-bit keys.</td></tr>
<tr><td>FCS_COP.1/SigGen</td><td>The TOE provides cryptographic signature services using RSA and ECDSA with key sizes (modulus) of 2048 and 3072 bits, and 256, 384, and 521 bits, respectively. For RSA, the key size is configurable down to 1024, but only 2048 or greater key size is permitted in the evaluated configuration. IKE/IPsec supports both ECDSA and RSA digital signature. SSH and trusted update only support RSA digital signature. The key generation is also tested as part of the signature generation and verification functions.</td></tr>
<tr><td>FCS_COP.1/Hash, FCS_COP.1/ KeyedHash</td><td>The TOE provides cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512, and keyed-hash message authentication using HMAC-SHA-1 (160-bit), HMAC-SHA-256 (256-bit), HMAC-SHA-384 (384-bit), and HMAC-SHA-512 (512-bit) with block size of 64 bytes (HMAC-SHA-1 and HMAC-SHA-256) and 128 bytes (HMAC-SHA-384 and HMAC-SHA-512).</td></tr>
<tr><td>FCS_RBG_EXT.1</td><td>In the TOE cryptographic functions are used to establish TLS, HTTPS, IPsec, and SSH sessions, for IPsec traffic and authentication keys, and for IKE authentication and encryption keys.</td></tr>
</table>

TOE SFRs	How the SFR is Satisfied
	Random number generation in the Firepower 2100 series hardware appliances uses a hardware-based NIST SP-800-90 Hash_DRBG with SHA-512, and a firmware-based NIST SP 800-90 CTR_DRBG with AES-256. The DRBG is seeded by an entropy source (Cavium CN70XX/CN71XX OCTEON® III) that is of 312-bit value as described in the proprietary Entropy Design document. Proprietary information on the entropy source is provided in the entropy design documentation. In the Firepower 1000 series appliances, the DRBG is seeded by a NIST SP-800-90 CTR_DRBG entropy source (ACT2Lite (ACT2) chip) that is of 323-bit value as described in the proprietary Entropy Design document
FCS_HTTPS_EXT.1 FCS_TLSC_EXT.1 FCS_TLSC_EXT.2 FCS_TLSS_EXT.1	The TOE implements HTTP over TLS (HTTPS) to support remote administration using ASDM and TLS client to support secure syslog connection. A remote administrator can connect over HTTPS to the TOE with their web browser and load the ASDM software. The TOE supports TLS v1.2 and TLSv1.1[2] connections with any of the following ciphersuites: • TLS_DHE_RSA_WITH_AES_128_CBC_SHA • TLS_DHE_RSA_WITH_AES_256_CBC_SHA • TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 (TLS v1.2 only) • TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 (TLS v1.2 only) • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 (TLS v1.2 only) • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 (TLS v1.2 only) When the TOE acts as a TLS client, the administrators can specify the reference-identity using the following command: asa(config)#**crypto ca reference-identity** *reference-identity-name* Followed by one or more of the values (where cn-id would be used to specify the FQDN or DN): cn-id *value* dns-id *value* uri-id *value* For example, ciscoasa(config)# crypto ca reference-identity syslogServer ciscoasa(config-ca-ref-identity)# cn-id syslog.cisco.com

[2] TLS version support is configurable by administrator. Connections not supporting the configured TLS version will not be established.

TOE SFRs	How the SFR is Satisfied
	To configure the syslog server certification[3] verification, use this syntax: logging host *interface_name syslog_ip* [tcp/*port* | udp/*port*] [format emblem] [secure [reference-identity *reference_identity_name*]] [permit-hostdown] For example, ciscoasa(config)# logging host outside 10.1.2.123 tcp/6514 secure **reference- identity** syslogServer When ASA validates a certificate against a reference-identity configuration: • uri-id is matched literally • cn-id and dns-id support wildcards NIST curves - secp256r1, secp384r1 and secp521r1, are supported by default but mutual authentication must be configured with the client-side X.509v3 certificate. The TOE can be configured to specify which TLS versions are supported using asa(config)#**ssl client-version** *{~~tlsv1 |~~ tlsv1.1 | tlsv1.2}* asa(config)#**ssl server-version** *{~~tlsv1 |~~ tlsv1.1 | tlsv1.2}* The key agreement parameters of the server key exchange message are specified in the RFC 5246 (section 7.4.3) for TLSv1.2 and RFC 4346 (section 7.4.3) for TLSv1.1. The TOE performs key establishment for TLS using Diffie-Hellman parameters with size 2048 bits and 3072 bits and ECDHE curves -secp256r1, secp384r1 and secp521r1. [DH group 14, 15, 19, 20 and 24]. TLS session resumption is supported for the TLS connections of the TOE based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2). The TOE tracks the negotiated sessions based on session IDs and when the TLS client reconnects to the server based on session IDs, it can look up the session keys to resume the encrypted session.
FCS_IPSEC_EXT.1	The IPsec implementation provides both VPN peer-to-peer (i.e., site-to-site) and VPN client to TOE (i.e., remote access) capabilities. The VPN peer-to-peer tunnel allows for example the TOE and another TOE to establish an IPsec tunnel to secure the passing of user data. Another configuration in the peer-to-peer configuration is to have the TOE be set up with an IPsec tunnel with a VPN peer to secure the session between the TOE and syslog server. The VPN client to TOE configuration would be where a remote VPN client connects into the TOE in order to gain access to an authorized private network. Authenticating with the TOE would give the VPN client a secure IPsec tunnel to connect over the internet into their private network. The TOE's implementation of IPsec relies mainly on the Octeon III crypto chip, but the RSA key generation and verification and ECDSA key generation and verification is done via the FOM 7.2sp implementation. The TOE implements IPsec to provide both certificates and pre-shared key-based authentication and encryption services to prevent unauthorized viewing or

[3] Certificate pinning is not supported. In addition, IP address is not supported in the ID.

TOE SFRs	How the SFR is Satisfied
	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, ECDSA algorithm with X.509v3 certificates, or pre-shared keys. 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 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 peer authentication parameters, either signature based or pre-shared key based), • The establishment of additional Security Associations to protect packets 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 The TOE implements IPsec using the ESP protocol as defined by RFC 4303, using the cryptographic algorithms AES-CBC-128, AES-CBC-256, AES-GCM-128 and AES-GCM-256 (both specified by RFCs 3602 and 4106) along with SHA-based HMAC algorithms (HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512), and using IKEv2, as specified for FCS_IPSEC_EXT.1.5, 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). IKEv2 SA can be limited by time only. IKEv2 Child SA can be limited by time or number of kilobytes (The valid range in kilobytes is 10 to 2,147,483,647 (10KB to 2TB). The time is in number of seconds. Administrators can require use of main mode by configuring the mode for each IPsec tunnel, as in the following examples: asa(config)#**crypto map** *map-name seq-num* **set ikev2 phase1-mode main** asa(config)# crypto ipsec security-association lifetime {seconds *seconds* | kilobytes *kilobytes*} asa(config-ikev2-policy)# **lifetime seconds** *seconds* In the evaluated configuration, use of “confidentiality only” (i.e. using ESP without authentication) for IPsec connections is prohibited. The TOE allows the

TOE SFRs	How the SFR is Satisfied
	administrator to define the IPsec proposal for any IPsec connection to use specific encryption methods and authentication methods as in the following examples: asa(config)#**crypto ipsec ikev2 ipsec-proposal** *proposal tag proposal_name* asa(config-ipsec-proposal)#**protocol esp encryption** {aes | ~~aes-192~~ | aes-256 | aes-gcm | ~~aes-gcm-192~~ | aes-gcm-256 ~~| aes-gmac | aes-gmac-192 | aes-gmac-256~~} asa(config-ipsec-proposal)#**protocol esp integrity** {sha-1 | sha-256 | sha-384 | sha-512 | null} **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), 20 (384-bit Random ECP) and use the RSA and ECDSA algorithms for Peer Authentication. The following commands are used to specify the DH Group and other algorithms for SAs: asa(config)#**crypto ikev2 policy** *priority policy_index* asa(config-ikev2-policy)#**encryption** [~~null | des | 3des |~~ aes | ~~aes-192~~ | aes-256 | aes-gcm | ~~aes-gcm-192~~ | aes-gcm-256] asa(config-ikev2-policy)#**integrity [**~~md5 |~~ sha | sha256 | sha384 | sha512**]** asa(config-ikev2-policy)#**group** {14 | 19 | 20} asa(config-ikev2-policy)#**prf** {sha | sha256 | sha384 | sha512} The secret 'x' (nonce) 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 random number generator used to generate the nonces meets the requirements specified in FCS_RBG_EXT.1 for random bit generation. 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 **crypto ipsec ikev2 sa-strength-enforcement** command. 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). To configure an IKEv2 connection to use a RSA or ECDSA signature: asa(config)#**tunnel-group** *name* **ipsec-attributes** asa(config-tunnel-ipsec)#**ikev2** {local-authentication | remote-authentication} **certificate** *trustpoint* To define rules for matching the DN or FQDN of the IPsec peer certificate, use the **crypto ca certificate map** command to create a certificate map with the mapping rules, then associate certificate map with the tunnel-group. For example, a DN or FQDN can be specified by defining a rule for "subject-name" with attribute tag "san" to define a SAN (Subject Alternate name) mapping rule. Use one of the

TOE SFRs	How the SFR is Satisfied
	operators “co” (contains), “eq” (equals), “nc” (does not contain), or “ne” (is not equal to) to define the mapping rule for the specified string. asa(config)#**crypto ca certificate map {** *sequence-number* | *map-name sequence-number*} ciscoasa(ca-certificate-map)# **subject-name** [**attr** *tag* **eq | ne |co | nc** *string*] Pre-shared keys can be configured in TOE for IPsec connection authentication. However, pre-shared keys are only supported when using IKEv2 for peer-to-peer VPNs. The text-based pre-shared keys can be composed of any combination of upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “?”, space “ “, tilde~, hyphen-, underscore_, plus+, equal=, curly-brackets{}, square-brackets[], vertical-bar(pipe)|, forward-slash/, back-slash\, colon:, semi-colon;, double-quote“, single-quote‘, angle-brackets<>, comma,, and period.. The text-based pre-shared keys can be 1-128 characters in length and is conditioned by a“prf” (pseudo-random function) configurable by the administrator. The bit-based pre-shared keys can be entered as HEX value as well. When using pre-shared keys for authentication, the IPsec endpoints must both be configured to use the same key. To configure an IKEv2 connection to use a pre-shared key: asa(config)#**tunnel-group** *name* **ipsec-attributes** asa(config-tunnel-ipsec)#**ikev2** {local-authentication | remote-authentication} **pre-shared-key hex** *key-value* 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.
FCS_SSHS_EXT.1	The TOE implements SSHv2 (telnet is disabled in the evaluated configuration). SSHv2 sessions are limited to a configurable session timeout period of 120 seconds, a configurable max number of failed authentication attempts (default is 3). An SSH connection is rekeyed after 60 minutes of connection time or 1 GB of data traffic (audit log is generated to indicate successful rekey), whichever threshold is met first. SSH connections will be dropped if the TOE receives a packet larger than 65,535 bytes. The TOE’s implementation of SSHv2 supports: • Public key algorithm RSA for signing and verification; Public key-based authentication for administrative users. The TOE supports SSH public-key

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td></td><td>authentication using ssh-rsa and supports password-based authentication. 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.

Password-based authentication for administrative users. Password-based authentication can be performed on the serial console or SSH.
Public-key based authentication implementation uses ssh-rsa as its public key algorithm(s) and rejects all other public key algorithms
Encryption algorithms, AES-CBC-128, AES-CBC-256 to ensure confidentiality of the session;
Hashing algorithm hmac-sha1 and hmac-sha2-256 to ensure the integrity of the session.
Requiring use of DH group 14 (configurable using the following command when enabling SSHv2:

asa(config)#ssh key-exchange dh-group14 {ip_address mask | ipv6_address/prefix} interface</td></tr>
<tr><td>FIA_AFL.1</td><td>The TOE provides the privileged administrator the ability to specify the maximum number of unsuccessful authentication attempts (between 1 and 16) before privileged administrator or non-privileged administrator is locked out.

When a privileged administrator or non-privileged administrator attempting to login reaches the administratively set maximum number of failed authentication attempts, the user will not be granted access to the administrative functionality of the TOE until a privileged administrator resets the user's number of failed login attempts (i.e., unlocks) through the administrative CLI (local access is permitted).</td></tr>
<tr><td>FIA_PMG_EXT.1</td><td>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 from this list - “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “ ” ‘ ` (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).

Minimum password length is settable by the Authorized Administrator, and support passwords of 3 to 127 characters. 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.</td></tr>
<tr><td>FIA_UIA_EXT.1</td><td>The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed. Administrative access to the TOE is facilitated</td></tr>
</table>

TOE SFRs	How the SFR is Satisfied
	through the TOE's CLI, and through the GUI (ASDM). The TOE mediates all administrative actions through the CLI and GUI. Once a potential administrative user attempts to access an administrative interface either locally or remotely, the TOE prompts the user for a user name and password. 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 also supports authentication via SSH. The administrators can login to the TOE using SSH keys which are provided during the SSH connection request. 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.
FIA_UAU_EXT.2	The TOE provides a local password based authentication mechanism as well as RADIUS and TACACS+ authentication. The administrator authentication policies include authentication to the local user database or redirection to a remote authentication server. Interfaces can be configured to try one or more remote authentication servers, and then fall back to the local user database if the remote authentication servers are inaccessible. The TOE can invoke an external authentication server to provide a single-use authentication mechanism by forwarding the authentication requests to the external authentication server (when configured by the TOE to provide single-use authentication). The process for authentication is the same for administrative access whether administration is occurring via a directly connected console cable or remotely via SSHv2 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.
FIA_UAU.7	When a user enters their password at the local console, the TOE displays only '*' characters so that the user password is obscured. For remote session authentication, the TOE does not echo any characters as they are entered.
FIA_X509_EXT.1/Rev FIA_X509_EXT.2 FIA_X509_EXT.3	The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for TLS and IPsec connections. Public key infrastructure (PKI) credentials, such as private keys and certificates are stored in a specific location, such as NVRAM and flash memory. 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

TOE SFRs	How the SFR is Satisfied
	session establishment while CA certificate is checked at both. The TOE conforms to standard RFC 5280 for certificate and path validation. The TOE can generate a RSA or ECDSA key pair that can be embedded in a Certificate Signing Request (CSR) created by the TOE. The key pair can be generated with the following command: asa(config)#**crypto key generate** [rsa [general-keys | label <name> | modules [~~512 | 768 | 1024 |~~ 2048 | 3072 | ~~4096~~] | noconfirm | usage-keys] | ecdsa [label <name> | elliptic-curve [256 | 384 | 521] | noconfirm]] 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). Both OCSP and CRL are configurable and may be used for certificate revocation check when the TOE is validating server certificates when initiating outbound TLS connections to syslog and AAA servers and for IPsec connections. 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 trustpoint. In all use cases (whether using CRL or OCSP) if the connection to determine the certificate validity cannot be established, the TOE will not accept the certificate. The administrators can configure a trustpoint and associate it with a crypto map. This will tell the TOE which certificate(s) to use during the validation process. When the TOE cannot establish a connection for the validity check (e.g., CRL checking), the trusted channel is not established. For more information, please refer to the Preparative Procedures & Operational User Guide for the Common Criteria Certified Configuration.
FMT_MOF.1/ManualUpdate	The TOE restricts the ability to enable of the security functions of the TOE to a Security Administrator. The TOE provides the ability for Security Administrators to enable or disable service and features, and access TOE data, such as audit data, configuration data, security attributes, information flow rules, and session thresholds. The TOE also restricts the ability to manage the cryptographic keys to just the Security Administrators.
FMT_MTD.1/CoreData	The TOE provides the ability for authorized administrators to access TOE data, such as audit data, configuration data, security attributes, routing tables, and session thresholds. The TOE also restricts access to TSF data so that no manipulation can be performed by non-administrators. Each of the predefined and administratively configured privilege level has default set of permissions that will grant them access to the TOE data, though with some privilege levels, the access is limited. The TOE performs role-based authorization, using TOE platform authorization mechanisms, to grant access to the semi-privileged and privileged levels. For the purposes of this evaluation, the privileged level is equivalent to full administrative access to the CLI or GUI, and equivalent to privilege level 15. The term "authorized administrator" or "Security Administrator" is used in this ST to refer to any user

TOE SFRs	How the SFR is Satisfied
	which has been assigned to a privilege level that is permitted to perform the relevant action.
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/VPN[VPN]	The TOE is configured to restrict the ability to enter privileged configuration mode to level 15 users (the authorized administrator) once AAA authorizations has been enabled. Privileged configuration (EXEC) mode is where the commands are available to modify user attributes (‘username’ and ‘password’ commands), operation of the TOE (‘reload’), authentication functions (‘aaa’ commands'), audit trail management (‘logging’ commands), communication with authorized external IT entities (‘access list’ commands), information flow rules (‘access list’ commands), modify the timestamp (‘clock’ commands), specify limits for authentication failures (‘aaa local authentication lockout’), etc. These commands are not available outside of this mode. Communications with external IT entities, include the host machine for ASDM. This is configured through the use of ‘https’ commands that enable communication with the host and limit the IP addresses from which communication is accepted. Note that the TOE does not provide services (other than connecting using HTTPS or SSH, and establishment of VPNs) prior to authentication so there are no applicable commends. There are specific commands for the configuration of cryptographic services. Trusted updates to the product can be verified using cryptographic digital signature. The ASDM uses the same privileges that the user would have at the CLI to determine access to administrative functions in the ASDM GUI. All administrative configurations are done through the ‘Configuration’ page. The management functions specified in the FMT_SMF.1, FMT_SMF.1/VPN[VPN] and FMT_SMF.1/FFW[FW] are available to the Security administrators through all three administrative interfaces.
FMT_SMR.2	The TOE supports multiple levels of administrators, the highest of which is a privilege 15. In this evaluation, privilege 15 would be the equivalent of the authorized administrator with full read-write access. Multiple level 15 administrators with individual usernames can be created. Through the CLI the ‘username’ command is used to maintain, create, and delete users. Through ASDM this is done on the ‘Configuration > Device Management > Users/AAA > User Accounts’ page. Usernames defined within the local user database are distinguished based on their privilege level (0-15) and the service-type attribute assigned to the username, which by default is “admin”, allowing the username to authenticate (with valid password) to admin interfaces. 'aaa authentication ssh console LOCAL' can be used to set the TOE to authenticate SSH users against the local database. 'aaa authorization exec' can be used to require re-authentication of users before they can get to EXEC mode.

TOE SFRs	How the SFR is Satisfied

	The TOE also supports creating of VPN User accounts, which cannot login locally to the TOE, but can only authenticate VPN sessions initiated from VPN Clients. VPN users are accounts with privilege level 0, and/or with their service-type attribute set to "remote-access".
	
	When command authorization has been enabled the default sets of privileges take effect at certain levels, and the levels become customizable.
	
	• When "aaa authorization command LOCAL" has NOT been applied to the config:
	    o All usernames with level 2 and higher have the same full read-write access as if they had level 15 once their interactive session (CLI or ASDM) is effectively at level 2 or higher.
	    o Usernames with privilege levels 1 and higher can login to the CLI, and "enable" to their max privilege level (the level assigned to their username).
	    o Usernames with privilege levels 2-14 can login to ASDM, and have full read-write access.
	    o Privilege levels cannot be customized.
	• When "aaa authorization command LOCAL" has been applied to the config:
	    o Default command authorizations for privilege levels 3 and 5 take effect, where level 3 provides "Monitor Only" privileges, levels 4 and higher inherit privileges from level 3, level 5 provides "Read Only" privileges (a superset of Monitor Only privileges), and levels 6-14 inherit privileges from level 5.
	    o Privilege levels (including levels 3 and 5) can be customized from the default to add/remove specific privileges.
	
	To display the set of privileges assigned to levels 3 or 5 (or any other privilege level), use "show running-config all privilege all", which shows all the default configuration settings that are not shown in the output of "show running-config all".

FPT_SKP_EXT.1	The TOE stores all private keys in a secure directory (an 'opaque' virtual filesystem in flash memory called "system:") that is not readily accessible to administrators. Pre-shared and symmetric keys are never visible in plaintext form, they're either stored and displayed in encrypted form (AES encrypted), or are stored in a non-encrypted form and masked when showing the configuration via administrative interfaces (CLI or GUI).
FPT_APW_EXT.1	All TOE administrative passwords are stored as PBKDF2 hashes. When an administrator sets a password via CLI or ASDM (e.g. using the "username" command via CLI), the TOE creates a PBKDF2 hash when it saves the password to the configuration. When an administrator views the stored configuration via CLI or ASDM (e.g. using the "show running-config" command via CLI), the configuration does not display the actual password, it shows the hashed password followed by the "pbkdf2" keyword.

TOE SFRs	How the SFR is Satisfied
FPT_STM_EXT.1	The TOE provides a source of date and time information for the firewall, used in audit timestamps, in validating service requests, and for tracking time-based actions related to session management including timeouts for inactive administrative sessions (FTA_SSL_EXT.*), and renegotiating SAs for IPsec tunnels (FCS_IPSEC_EXT.1). This function can only be accessed from within the configuration exec mode via the privileged mode of operation of the firewall. The clock function is reliant on the system clock provided by the underlying hardware.
FPT_TST_EXT.1	The TOE runs a suite of self-tests during initial start-up (power-on-self-tests or POST) to verify its correct operation. FIPS mode must be enabled in the evaluated configuration. When FIPS mode is enabled on the TOE, 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 functions, signature hashing functions, and random number generation. The firmware integrity testing verifies the image has not been tampered with or corrupted. If any of these self-tests fails, the TOE will cease operation. For more details, please see FPT_FLS.1.
FPT_TUD_EXT.1	The TOE has specific versions that can be queried by an administrator. The administrator can determine the current executing version of the software. When updates are made available by Cisco, an administrator can obtain and manually install those updates. Digital signatures 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 TOE. The update process will fail if the digital signature verification process fails. Instructions on how to perform verification and update are provided in the Preparative Procedures & Operational User Guide for the Common Criteria Certified Configuration. The TOE is performing the integrity verification functions related to this SFR.
FTA_SSL_EXT.1	An administrator can configure maximum inactivity times for remote SSH and TLS 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.
FTA_SSL.3	
FTA_SSL.4	An administrator is able to exit out of both local and remote administrative sessions, effectively terminating the session so it cannot be re-used and will require authentication to establish a new session.

TOE SFRs	How the SFR is Satisfied
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 and HTTPS).
FTP_ITC.1	The TOE uses IPsec 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 message over IPsec and/or TLS. • Connections to authentication servers (AAA servers) can be protected via IPsec tunnels. Connections with AAA servers (via RADIUS/TACACS+) can be configured for authentication of TOE administrators. • Connections to VPN peers can be initiated from the TOE using IPsec. In addition the TOE can establish secure VPN tunnels with IPsec VPN clients. Note that the remote VPN client is in the operational environment.
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 TOE also supports tunneling the remote administration (ASDM and/or SSH connections) in IPsec VPN tunnels (peer-to-peer, or remote VPN client).
Security Functional Requirements Drawn from mod_cpp_fw_v1.4e	
FDP_RIP.2[FW]	The TOE ensures that packets transmitted through the TOE do not contain residual information from previous packets. Packets that are not the required length use zeros for padding. Residual data is never transmitted from the TOE. Packet handling within memory buffers ensures new packets cannot contain portions of previous packets. This applies to data plane traffic and even administrative session traffic.
FFW_RUL_EXT.1.1[FW] FFW_RUL_EXT.1.2[FW]	The TOE provides stateful traffic filtering of IPv4 and IPv6 network traffic. Administratively-defined traffic filter rules (access-lists) 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 7.1.

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td></td><td>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 2460 (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 critical subcomponent 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 subcomponent 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>The TOE supports filtering of the following protocols and enforces proper session establishment, management, and termination as defined in each protocol’s RFC including proper use of:
• 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 2460 (IPv6);
• Source and destination addresses, 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);
• Source and destination addresses, 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).</td></tr>
<tr><td>FFW_RUL_EXT.1.3[FW],
FFW_RUL_EXT.1.4[FW]</td><td>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 of any permitted or denied traffic (in this case, attempt to establish a session). 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 with the “access-group” command, e.g. “access-group sample-acl in interface outside”. Interfaces can be referred to</td></tr>
</table>

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td></td><td>by their identifier (e.g. GigabitEthernet 0/1), or by a name if named using the “nameif” command e.g.:

asa(config)# interface gigabitethernet0/1

asa(config-if)# nameif inside

The interface types that can be assigned to an access-group are:
Physical interfaces
Ethernet
GigabitEthernet
TenGigabitEthernet
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:
Physical interfaces = Disabled
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.

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.</td></tr>
<tr><td>FFW_RUL_EXT.1.5[FW]</td><td>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,</td></tr>
</table>

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td></td><td>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</td></tr>
<tr><td>FFW_RUL_EXT.1.6[FW],
FFW_RUL_EXT.1.7[FW]</td><td>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 the packets/frames that were dropped.

The TOE’s can be used to deny and log traffic by defining policies with the “ip audit name” command, specifying the “drop” action, and applying the policy or policies to each enabled interface. Each signature has been classified as either “informational”, or “attack”. Using the “info” and “attack” keywords in the “ip audit name” command defines the action the TOE will take for each signature classification.

asa(config)# ip audit name name {info | attack} [action [alarm] [drop] [reset]]
asa(config)# ip audit interface interface_name policy_name

Example:

asa(config)# ip audit name ccpolicy1 attack action alarm reset
asa(config)# ip audit name ccpolicy2 info action alarm reset
asa(config)# ip audit interface outside ccpolicy1
asa(config)# ip audit interface inside ccpolicy2

Specifying the “alarm” action in addition to the “drop” action will result in generating an audit message when the signature is detected. Messages 400000 through 400051 are Cisco Intrusion Prevention Service signature messages, and have this format:</td></tr>
</table>

TOE SFRs	How the SFR is Satisfied
	%ASA-4-4000*nn*: IPS:*number string* from *IP_address* to *IP_address* on interface *interface_name* The following traffic will be denied by the TOE, and audit messages will be generated as indicated: *1.* packets which are invalid fragments, including IP fragment attack 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). The output of the "show fragment" command displays the count (the 'fail' value) of packets that failed reassembly on each interface. The command "clear fragment statistics [*interface_name*]" can be used to reset those counters. *2.* fragmented IP packets which cannot be re-assembled completely 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). The output of the "show fragment" command displays the current fragment reassembly thresholds for each interface, as well as the count (the 'overflow' value) of fragments per interface that have been dropped, and the count (the 'fail' value) of packets that failed reassembly due to an 'overflow' of one of the configured fragment reassembly thresholds. *3.* packets where the source address of the network packet is equal to the address of the network interface where the network packet was received; %ASA-2-106016: Deny IP spoof from (*IP_address*) to *IP_address* on interface *interface_name.* *4.* 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; %ASA-2-106016: Deny IP spoof from (*IP_address*) to *IP_address* on interface *interface_name.* This next message appears when Unicast RPF has been enabled with the **ip verify reverse-path** command. %ASA-1-106021: Deny *protocol* reverse path check from *source_address* to *dest_address* on interface *interface_name* This next message appears when a packet matching a connection arrived on a different interface from the interface on which the connection began, and the **ip verify reverse-path** command is not configured. %ASA-1-106022: Deny *protocol* connection spoof from *source_address* to *dest_address* on interface *interface_name* *5.* packets where the source address of the network packet is defined as being on a broadcast network;

TOE SFRs	How the SFR is Satisfied
	%ASA-2-106016: Deny IP spoof from (*IP_address*) to *IP_address* on interface *interface_name.* *6.* packets where the source address of the network packet is defined as being on a multicast network; %ASA-4-106023: Deny *protocol* src [*interface_name*:*source_address/source_port*] dst i*nterface_name*:*dest_address/dest_port* [type {*string*}, code {*code*}] by access_group *acl_ID* The following message will be generated when the rules listed below are configured without the "log" option. %ASA-4-106100: access-list *acl_ID* denied *protocol* *interface_name*/*source_address*(*source_port*) - *interface_name*/*dest_address*(*dest_port*) hit-cnt *number* ({first hit | *number*-secondinterval}) hash codes The following message will be generated when these rules are configured with the "log" option: asa(config)#**object-group network** *grp_name* asa(config-network-object-group)#**network-object** 224.0.0.0 255.0.0.0 #IPv4 multicast asa(config-network-object-group)#**network-object** FF00::/8 #IPv6 multicast asa(config)#**access-list** *acl-name* **extended deny ip** *grp-name* **any [log]** asa(config)#**access-group in interface** *int-name* *7.* packets where the source address of the network packet is defined as being a loopback address; %ASA-2-106016: Deny IP spoof from (*IP_address*) to *IP_address* on interface *interface_name.* The following message will be generated when no ACL has been defined to explicitly deny this traffic. %ASA-4-106023: Deny *protocol* src [*interface_name*:*source_address/source_port*] dst i*nterface_name*:*dest_address/dest_port* [type {*string*}, code {*code*}] by access_group *acl_ID* The following message will be generated when the rules listed below are configured without the "log" option. %ASA-4-106100: access-list *acl_ID* denied *protocol* *interface_name*/*source_address*(*source_port*) - *interface_name*/*dest_address*(*dest_port*) hit-cnt *number* ({first hit | *number*-secondinterval}) hash codes The following message will be generated when these rules are configured with the "log" option: asa(config)#**object-group network** *grp_name* asa(config-network-object-group)#**network-object** 127.0.0.0 255.0.0.0 #IPv4 loopback asa(config-network-object-group)#**network-object** ::1/128 #IPv6 loopback asa(config)#**access-list** *acl-name* **extended deny ip** *grp-name* **any [log]** asa(config)#**access-group in interface** *int-name* *8.* packets where the source or destination address of the network packet is a link-local address;

TOE SFRs	How the SFR is Satisfied
	%ASA-2-106016: Deny IP spoof from (*IP_address*) to *IP_address* on interface *interface_name.* The following message will be generated when no ACL has been defined to explicitly deny this traffic. %ASA-4-106023: Deny *protocol* src [*interface_name*:*source_address/source_port*] dst i*nterface_name*:*dest_address/dest_port* [type {*string*}, code {*code*}] by access_group *acl_ID* The following message will be generated when the rules listed below are configured without the "log" option. %ASA-4-106100: access-list *acl_ID* denied *protocol* *interface_name*/*source_address*(*source_port*) - *interface_name*/*dest_address*(*dest_port*) hit-cnt *number* ({first hit | *number*-secondinterval}) hash codes The following message will be generated when these rules are configured with the "log" option: asa(config)#**object-group network** *grp_name* asa(config-network-object-group)#**network-object** 127.0.0.0 255.0.0.0 #IPv4 link-local asa(config-network-object-group)#**network-object** FE80::/10 #IPv6 link-local asa(config)#**access-list** *acl-name* **extended deny ip** *grp-name* **any [log]** asa(config)#**access-list** *acl-name* **extended deny ip any** *grp-name* **[log]** asa(config)#**access-group in interface** *int-name* *9.* 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" as specified in RFC 3513 for IPv6; %ASA-4-106023: Deny *protocol* src [*interface_name*:*source_address/source_port*] dst i*nterface_name*:*dest_address/dest_port* [type {*string*}, code {*code*}] by access_group *acl_ID* The following message will be generated when the rules listed below are configured without the "log" option. %ASA-4-106100: access-list *acl_ID* denied *protocol* *interface_name*/*source_address*(*source_port*) - *interface_name*/*dest_address*(*dest_port*) hit-cnt *number* ({first hit | *number*-secondinterval}) hash codes The following message will be generated when these rules are configured with the "log" option: asa(config)#**object-group network** *grp_name* asa(config-network-object-group)#**network-object** 192.0.0.0 255.0.0.0 #IPv4 reserved asa(config-network-object-group)#**network-object** 240.0.0.0 128.0.0.0 #IPv4 reserved asa(config)#**access-list** *acl-name* **extended deny ip** *grp-name* **any [log]** asa(config)#**access-list** *acl-name* **extended deny ip any** *grp-name* **[log]** asa(config)#**access-group in interface** *int-name* *10.* 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" as specified in RFC 3513 for IPv6; %ASA-4-106023: Deny *protocol* src [*interface_name*:*source_address/source_port*] dst i*nterface_name*:*dest_address/dest_port* [type {*string*}, code {*code*}] by access_group *acl_ID*

TOE SFRs	How the SFR is Satisfied
	The following message will be generated when the rules listed below are configured without the "log" option. %ASA-4-106100: access-list *acl_ID* denied *protocol* *interface_name*/*source_address*(*source_port*) - *interface_name*/*dest_address*(*dest_port*) hit-cnt *number* ({first hit | *number*-secondinterval}) hash codes The following message will be generated when these rules are configured with the "log" option: asa(config)#**object-group network** *grp_name* asa(config-network-object-group)#**network-object** :: #IPv6 unspecified asa(config-network-object-group)#**network-object** 0000::/8 #IPv6 reserved asa(config)#**access-list** *acl-name* **extended deny ip** *grp-name* **any [log]** asa(config)#**access-list** *acl-name* **extended deny ip any** *grp-name* **[log]** asa(config)#**access-group in interface** *int-name* *11.* Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; %ASA-6-106012: Deny IP from *IP_address* to *IP_address*, IP options *hex*. The following messages will be generated when configured as described above. %ASA-4-400001: IPS:1001 IP options-Record Packet Route from *IP_address* to *IP_address* on interface *interface_name* %ASA-4-400004: IPS:1004 IP options-Loose Source Route from *IP_address* to *IP_address* on interface *interface_name* %ASA-4-400006: IPS:1006 IP options-Strict Source Route from *IP_address* to *IP_address* on interface *interface_name* *12.* By default, TOE will also drop (and is capable of logging) a variety of other IP packets with invalid content including: • In routed mode when the TOE receives a through-the-box: 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 • ICMP error packets when the ICMP error messages are not related to any session already established in the TOE. • Network packets when the appliance is not able to find any established connection related to the frame embedded in the ICMPv6 error message. • Network packets when an ICMP echo request/reply packet was received with a malformed code (non-zero).
FFW_RUL_EXT.1.8[FW]	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

TOE SFRs	How the SFR is Satisfied
	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 changes the sequence in which the entries are compared to network traffic flows.
FFW_RUL_EXT.1.9[FW]	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]	TOE administrators can configure the maximum number of half-open TCP connections allowed using the "set connection embryonic-conn-max *0-65535*" in the service-policy command. 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, 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.

FFW_RUL_EXT.2[FW],
FMT_SMF.1/FFW[FW]
	The TOE supports dynamic establishment of secondary network sessions (e.g., FTP). 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.

TOE SFRs	How the SFR is Satisfied
Reproduced from mod_vpngw_v1.1	
FCS_CKM.1/IKE[VPN]	See FCS_CKM.1
FIA_PSK_EXT.1[VPN]	The TOE supports use of IKEv2 pre-shared keys for authentication of IPsec tunnels. Pre-shared keys can be entered as ASCII character strings, or HEX values. The text-based pre-shared keys can be composed of any combination of upper- and lower-case letters, numbers, and special characters. The TOE supports keys that are from 1 character in length up to 128 in length. The text-based pre-shared key is conditioned by one of the prf functions (HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 or HMAC-SHA-512) configured by the administrator.
FPF_RUL_EXT.1[VPN]	An authorized administrator can define the traffic that needs to be protected by configuring access lists (permit, deny, log) and applying these access lists to interfaces using access and crypto map sets. 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 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. 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. 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 16 IPv6 protocols: • Transport Layer Protocol 4 - IPv4 encapsulation

TOE SFRs	How the SFR is Satisfied
	• 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 • Transport Layer Protocol 132 - Stream Control Transmission Protocol Protocol 2 (IGMP) and Protocol 103 (PIM) are excluded as they are not routable and thus not forwarded by the TOE despite the TOE recognizing the 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. 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; 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.

TOE SFRs	How the SFR is Satisfied
	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.
FPT_FLS.1/SelfTest[VPN]	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 occurs within the TOE that results in the TOE ceasing operation, the TOE securely disables its interfaces to prevent the unintentional flow of any information to or from the TOE and reloads. So long as the failures persist, the TOE 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
FTA_SSL.3[VPN]	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 **vpn-idle-timeout** or **default-idle-timeout** commands in the VPN configuration.
FTA_TSE.1[VPN]	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. In addition, the **vpn-access-hours** command can be used to restrict access based on date and time.
FTA_VCM_EXT.1[VPN]	The TOE provides the option to assign the remotely connecting VPN client an internal network IP address. The **ip-local-pool** command can be used to define the range of IP and IPv6 addresses to be available for use.
FTP_ITC.1/VPN[VPN]	See FTP_ITC.1

7 Supplemental TOE Summary Specification Information

7.1 Tracking of Stateful Firewall Connections

7.1.1 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.

7.1.2 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 20: 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.
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 21: 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 a service module.
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 22: 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 TOE 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 TOE 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 TOE preallocates an outside back connection with this flag set. If the inside client attempts to connect back to the server, then the TOE 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 TOE preallocates an inside back connection with this flag set. If the outside server attempts to connect back to the client, then the TOE 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.
y	For clustering, identifies a backup owner flow. *(Clustering is not used in CC-certified configurations.)*
Y	For clustering, identifies a director flow. *(Clustering is not used in CC-certified configurations.)*
z	For clustering, identifies a forwarder flow. *(Clustering is not used in CC-certified configurations.)*
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 23: 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

7.1.3 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

7.2 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 memory (also known as flash memory). For all CSPs in DRAM, the CSPs are zeroized via API calls or power cycle. For all CSPs in NVRAM, the CSPs are zeroized via command that calls API.

Table 24: 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[4] within the two crypto modules (Cavium and CiscoSSL FOM), when module is shutdown, or reinitialized. Storage: DRAM Overwritten with: 0x00
Diffie Hellman Private 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.

[4] This function crypto_key_zeroize() overwrites the key buffer with zeroes and verifies that the overwrite was successful.

Critical Security Parameters (CSPs)	Zeroization Cause and Effect
	Storage: DRAM Overwritten with: 0x00
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 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
PRNG Seed Key	Seed keys are zeroized and overwritten with the generation of new seed. Storage: DRAM

Critical Security Parameters (CSPs)	Zeroization Cause and Effect
	Overwritten with: 0x00
RADIUS Secret	Zeroized using the following command: **# no radius-server key** Storage: NVRAM Overwritten with: 0x00
TACACS+ Secret	Zeroized using the following command: **# no tacacs-server key** Storage: NVRAM Overwritten with: 0x00
SSH Private Key	Automatically zeroized upon generation of a new key Storage: NVRAM Overwritten with: 0x00
SSH Session Key	Automatically zeroized when the SSH session is terminated. Storage: DRAM Overwritten with: 0x00
TLS Session Keys	Keys in RAM are zeroized upon rebooting the TOE. Storage: DRAM Overwritten with: 0x00
TLS Server Private Key	Zeroized when the HTTPS server is no longer in use. Storage: NVRAM Overwritten with: 0x00
All CSPs	Zeroized on-demand on all file systems via the “erase” command. Storage: NVRAM Overwritten with: 0x00

7.3 CAVP Certificate Equivalence

The TOE models, processors, and cryptographic modules included in the evaluation are shown in the following table. The cryptographic module used in all TOE platforms is the CiscoSSL FOM 7.2sp running on the Intel Xeon D-15XX processor (in FP 2100 series hardware models) and Intel Atom C3000 Series (in FP 1000 series hardware models). The table also includes the crypto hardware in each FP 2100 appliance model that provides cryptographic acceleration. The FP 1000 Series appliance models do not have crypto hardware and rely only on FOM 7.2sp.

Table 25: Model Processors

CPU Family	CPU Model (Microarchitecture)	FOM	Crypto Hardware	Physical Appliance Models
Intel Xeon D	Intel Xeon D-1526 (Broadwell)	CiscoSSL FOM 7.2sp	Cavium OCTEON III CN7230 MIPS64	FP2110
	Intel Xeon D-1528 (Broadwell)		Cavium OCTEON III CN7340 MIPS64	FP2120
	Intel Xeon D-1548 (Broadwell)		Cavium OCTEON III CN7350 MIPS64	FP2130
	Intel Xeon D-1577 (Broadwell)		Cavium OCTEON III CN7360 MIPS64	FP2140
Intel Atom C3000 Series	Intel Atom C3558 (Goldmont)		N/A	FP 1010
	Intel Atom C3858 (Goldmont)		N/A	FP1120
	Intel Atom C3958 (Goldmont)		N/A	FP1140, FP1150

Table 26: Algorithm Numbers

Algorithm	SFR	CiscoSSL FOM 7.2sp	Octeon III Family Crypto Engine Hardware (Only in FP 2100 Series models)
AES CBC 128/256 GCM 128/256	FCS_COP.1/DataEncryption	A2445	3301
RSA	FCS_COP.1/SigGen	A2445	1745 (SigGen/SigVer only)

(2048 and 3072 bits) Signature Gen & Verify Key Gen	FCS_CKM.1		
DSA (2048 and 3072 bits)	FCS_CKM.1	A2445	n/a
ECDSA curves P-256, P-384 and P-521 Signature Gen & Verify Key Gen and Verify	FCS_COP.1/SigGen FCS_CKM.1	A2445	n/a
Hashing SHA-1, SHA-256, SHA-384, SHA-512	FCS_COP.1/Hash	A2445	2737
Keyed Hash HMAC-SHA-1, HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	FCS_COP.1/KeyedHash	A2445	2095
DRBG CTR_DRBG Hash_DRBG	FCS_RBG_EXT.1	A2445 (CTR_DRBG)	819 (Hash_DRBG)
CVL KAS ECC/FCC	FCS_CKM.2	A2445	n/a

8 ANNEX A: REFERENCES

The following documentation was used to prepare this ST:

Table 27: 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 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