PacStar 451/453/455 Series with Cisco ASAv 9.12 Security Target

Version 0.4
March 18, 2021

Prepared for:

Pacific Star Communications, Inc (dba PacStar)

15055 SW Sequoia
Pkwy. #100, Portland, OR 97224

Prepared By:

www.gossamersec.com

LIST OF TABLES

1. Security Target Introduction

This section identifies the Security Target (ST) and Target of Evaluation (TOE) identification, ST conventions, ST conformance claims, and the ST organization. The TOE is PacStar 451/453/455 Series provided by Pacific Star Communications, Inc. (dba PacStar). The TOE is being evaluated as a Firewall and VPN Gateway

The Security Target contains the following additional sections:

- Conformance Claims (Section 2)
- Security Objectives (Section 3)
- Extended Components Definition (Section 4)
- Security Requirements (Section 5)
- TOE Summary Specification (Section 6)

Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
    - Iteration: allows a component to be used more than once with varying operations. In the ST, iteration is indicated by a parenthetical number placed at the end of the component. For example FDP_ACC.1(1) and FDP_ACC.1(2) indicate that the ST includes two iterations of the FDP_ACC.1 requirement.
    - Assignment: allows the specification of an identified parameter. Assignments are indicated using bold and are surrounded by brackets (e.g., [**assignment**]). Note that an assignment within a selection would be identified in italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
    - Selection: allows the specification of one or more elements from a list. Selections are indicated using bold italics and are surrounded by brackets (e.g., [***selection***]).
    - Refinement: allows the addition of details. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …").
- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.

1.1 Security Target Reference

ST Title –PacStar 451/453/455 Series with Cisco ASAv 9.12 Security Target

ST Version – Version 0.4

ST Date – March 18,2021

1.2 TOE Reference

TOE Identification –PacStar 451/453/455 Series with Cisco ASA Virtual Version 9.12

TOE Developer – Pacific Star Communications, Inc. (dba PacStar)

Evaluation Sponsor – Pacific Star Communications, Inc. (dba PacStar)

1.3 TOE Overview

The Target of Evaluation (TOE) is PacStar 451/453/455 Series running Adaptive Security Appliance Virtual (ASAv) Version 9.12.

The TOE is a purpose-built, firewall platform with VPN capabilities. The TOE includes the hardware models as defined in Table 1 of section 1.4

1.4 TOE Description

This section provides an overview and description of the TOE. The TOE is comprised of both software and hardware. The TOE hardware is PacStar 451/453/455 Series, while the software is comprised of the ASAv software image Release 9.12) running on ESXi 6.0 or 6.5.

The hardware models included in the scope are as follows. All models run the same ASAv image.

Hardware Model	Processor
PacStar 451 - Xeon D Family	Intel Xeon D 1559, Intel Xeon D 1539, Intel Xeon D 1577 (Broadwell)
PacStar 451 - Xeon E Family	Intel Xeon E-2254ML, Intel Xeon E-2276ME (Coffee Lake)
PacStar 453 - Xeon D Family	Intel Xeon D 1559, Intel Xeon D 1539, Intel Xeon D 1577 (Broadwell)
PacStar 455 - Xeon D Family	Intel Xeon D 1559, Intel Xeon D 1539, Intel Xeon D 1577 (Broadwell)

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 TOE's logical boundary surrounds the physical enclosure of the PacStar 451/453/455 Series chasses and the TOE's Operational Environment (OE) includes the following:

- VPN Peer (Operational Environment) or another instance of the TOE
- VPN Peer (Operational Environment) with Cisco VPN Client or AnyConnect Client
- Management Workstation (Operational Environment) with ASDM
- Remote Authentication Server (Operational Environment)
- Peer CA (Operational Environment)
- Syslog server (Operational Environment)

1.4.1 TOE Architecture

The TOE consists of hardware and software that provide connectivity and security services onto a single, secure device.

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

In addition to IP header information, the TOE mediates information flows based on 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-gateway1 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 transparent firewall with two interfaces connected to the same subnet when deployed in single context in transparent mode; or with one or more contexts connected to two or many IP subnets when configured in routed mode.

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

- TLS/HTTPS encrypted sessions.
- Rapid Configuration: in-line and drag-and-drop policy editing, auto complete, configuration wizards, appliance software upgrades, and online help;
- Powerful Diagnostics: Packet Tracer, log-policy correlation, packet capture, regular expression tester, and embedded log reference;
- Real-Time Monitoring: device, firewall, content security, real-time graphing; and tabulated metrics;
- Management Flexibility: A lightweight and secure design enables remote management of multiple security appliances

1.4.1.1 Supported non-TOE Hardware/ Software/ Firmware

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

Table 1: 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.
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

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

Component	Required	Usage/Purpose Description for TOE performance
		Note that that ASDM software is downloaded from the ASAv 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. This can be any AAA server that provides single-use authentication. 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.
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.
NTP Server	No	The TOE supports communication with an NTP server

1.4.1.2 Physical Boundaries

The physical boundary of the TOE is the appliance itself.

Hardware Model	Description
PacStar 451 - Xeon D Family	Up to 128 GB RAM, Dual SFP+ 10G ports, 2 GigE ports, single 2.5 inch or dual M.2 SATA SSD, 1 USB, one full size Display Port.
PacStar 451 - Xeon E Family	Up to 96 GB RAM and (5x GigE ethernet ports, mini Display Port, and 1x USB Port) or (3x GigE ethernet ports, 4x USB, and 1x full sized Display Port)
PacStar 453 - Xeon D Family	Up to 128 GB RAM, Dual SFP+ 10G ports, 2 GigE ports, dual 2.5 inch SATA SSDs, , 1

	USB, one full size Display Port. Includes one or more GPU options
PacStar 455 - Xeon D Family	Up to 128 GB RAM, Dual SFP+ 10G ports, 2 GigE ports, up to 8x 2.5 inch SATA SSDs, , 1 USB, one full size Display Port.

1.4.1.3 Logical Boundaries

This section summarizes the security functions provided by the PacStar 451/453/455 Series:

- Security audit
- Cryptographic support
- User data protection
- Firewall/Packet Filtering
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels

1.4.1.3.1 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.

1.4.1.3.2 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.

1.4.1.3.3 User data protection

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

1.4.1.3.4 Firewall/Packet 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.

1.4.1.3.5 Identification and authentication

The TOE performs two types of authentication: device-level authentication of the remote device (VPN peers) and user authentication for the authorized administrator of the TOE. 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 8-127 characters. The TOE also implements a lockout mechanism if the number of configured unsuccessful threshold 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 any AAA server (part of the IT Environment) for authentication of administrative users attempting to connect to the TOE.

1.4.1.3.6 Security management

The TOE provides secure administrative services for management of general TOE configuration and the security functionality provided by the TOE. All TOE administration occurs either through a secure SSHv2 or TLS/HTTPS session, or via a local console connection. 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; TOE configuration file storage and retrieval, 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.

1.4.1.3.7 Protection of the TSF

The TOE protects against interference and tampering by untrusted subjects by implementing identification, authentication, and access controls 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.

1.4.1.3.8 TOE access

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

1.4.1.3.9 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. 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.

1.4.2 TOE Documentation

Pacific Star Communications, Inc., (dba PacStar) PacStar 451/453/455 Series Administrator Guide, Version 0.1, March 18, 2021

2. Conformance Claims

This TOE is conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security functional components, Version 3.1, Revision 5, April 2017.
    - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security assurance components, Version 3.1, Revision 5, April 2017.
    - Part 3 Conformant
- Package Claims:
    - PP-Configuration for Network Devices, Stateful Traffic Filter Firewalls, and Virtual Private Network (VPN) Gateways, version 1.0, 6 March 2020 (CFG_NDcPP-FW-VPNGW_V1.0)
    - The PP-Configuration includes the following components:
        - collaborative Protection Profile for Network Devices, Version 2.1, 24 September 2018 (NDcPP21)
        - PP-Module for Virtual Private Network (VPN) Gateways, 1.0, 17 September 2019 (VPNGW10)
        - PP-Module for Stateful Traffic Filter Firewalls, 1.3, 27 September 2019 (STFFW13)
    - The TOE and ST are conformant with the Protection Profiles as listed above. The following NIAP Technical Decisions (TD) have also been applied:

TD #	TD Name	Protection Profiles	Applied to this TOE
TD0572	NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	CPP_ND_V2.1	FTP_ITC.1
TD0571	NiT Technical Decision for Guidance on how to handle FIA_AFL.1	CPP_ND_V2.1	FIA_UAU.1, FIA_PMG_EXT.1
TD0570	NiT Technical Decision for Clarification about FIA_AFL.1	CPP_ND_V2.1	FIA_AFL.1
TD0565	NiT Technical Decision for Firewall IPv4 & IPv6 testing by default	STFFW13	FFW_RUL_EXT.1
TD0549	Consistency of Security Problem Definition update for VPNGW10 and MOD_VPNGW_v1.1	VPNGW10	A.CONNECTIONS
TD0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	NDcPP21	AVA_VAN.1
TD0545	NIT Technical Decision for Conflicting FW rules cannot be configured (extension of RfI#201837)	STFFW13	FFW_RUL_EXT.1.8[FW]
TD0538	NIT Technical Decision for Outdated link to allowed-with list	NDcPP21	Section 2
TD0536	NIT Technical Decision for Update Verification Inconsistency	NDcPP21	FPT_TUD_EXT.1.3
TD0535	NIT Technical Decision for Clarification about digital signature algorithms for FTP_TUD.1	NDcPP21	FPT_TUD_EXT.1
TD0533	NIT Technical Decision for FTP_ITC.1 with signed downloads	NDcPP21	FTP_ITC.1
TD0532	NIT Technical Decision for Use of seeds with higher entropy	NDcPP21	FCS_RBG_EXT.1.2

TD0531	NIT Technical Decision for Challenge-Response for Authentication	NDcPP21	FCS_SSHS_EXT.1
TD0530	*NIT Technical Decision for FCS_TLSC_EXT.1.1 5e test clarification*	*NDcPP21*	*Not applied because FCS_TLSC_EXT.1.1 is not claimed.*
TD0529	NIT Technical Decision for OCSP and Authority Information Access extension	NDcPP21	FIA_X509_EXT.1/Rev , FIA_X509_EXT.2
TD0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	NDcPP21	FCS_NTP_EXT.1
TD0520	VPN Gateway SFR Rationale	VPNGW10	SFR-Objectives Rationale
TD0511	VPN GW Conformance Claim to allow for a PP-Module	VPNGW10	Section 2
TD0484	NIT Technical Decision for Interactive sessions in FTA_SSL_EXT.1 & FTA_SSL.3	NDcPP21	FTA_SSL_EXT.1, FTA_SSL.3
TD0483	NIT Technical Decision for Applicability of FPT_APW_EXT.1	NDcPP21	FPT_APW_EXT.1
TD0482	NIT Technical Decision for Identification of usage of cryptographic schemes	NDcPP21	FCS_CKM.1, FCS_CKM.2
TD0481	NIT Technical Decision for FCS_(D)TLSC_EXT.X.2 IP addresses in reference identifiers.	NDcPP21	FCS_TLSC_EXT.2
TD0480	NIT Technical Decision for Granularity of audit events	NDcPP21	FAU_GEN.1
TD0478	NIT Technical Decision for Application Notes for FIA_X509_EXT.1 iterations	NDcPP21	FIA_X509_EXT.1/Rev
TD0477	NIT Technical Decision for Clarifying FPT_TUD_EXT.1 Trusted Update	NDcPP21	FPT_TUD_EXT.1
TD0475	NIT Technical Decision for Separate traffic consideration for SSH rekey	NDcPP21	FCS_SSHS_EXT.1
TD0453	*NIT Technical Decision for Clarify authentication methods SSH clients can use to authenticate SSH se*	*NDcPP21*	*Not applied because this ST does not include FCS_SSHC_EXT.1*
TD0451	*NIT Technical Decision for ITT Comm UUID Reference Identifier*	*NDcPP21*	*Not applied because this ST does not include FPT_ITT*
TD0450	NIT Technical Decision for RSA-based ciphers and the Server Key Exchange message	NDcPP21	FCS_TLSS_EXT.*.3
TD0447	NIT Technical Decision for Using 'diffie-hellman-group-exchange-sha256' in FCS_SSHC/S_EXT.1.7	NDcPP21	FCS_SSHS_EXT.1.7
TD0425	NIT Technical Decision for Cut-and-paste Error for Guidance AA	NDcPP21	FTA_SSL.3
TD0424	NIT Technical Decision for NDcPP v2.1 Clarification - FCS_SSHC/S_EXT1.5	NDcPP21	FCS_SSHS_EXT.1.5
TD0423	NIT Technical Decision for Clarification about application of RfI#201726rev2	NDcPP21	FTP_ITC.1, FTP_TRP.1/Admin
TD0412	NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy	NDcPP21	FCS_SSHS_EXT.1.5

TD0411	*NIT Technical Decision for FCS_SSHC_EXT.1.5, Test 1 - Server and client side seem to be confused*	*NDcPP21*	*Not applied because this ST does not include FCS_SSHC_EXT.1*
TD0410	NIT technical decision for Redundant assurance activities associated with FAU_GEN.1	NDcPP21	FAU_GEN.1
TD0409	NIT decision for Applicability of FIA_AFL.1 to key-based SSH authentication	NDcPP21	FIA_AFL.1
TD0408	NIT Technical Decision for local vs. remote administrator accounts	NDcPP21	FIA_UAU_EXT.2.1, FIA_AFL.1.1, FIA_AFL.1.2
TD0407	*NIT Technical Decision for handling Certification of Cloud Deployments*	*NDcPP21*	*Not applied because this is specific to cloud deployments*
TD0402	NIT Technical Decision for RSA-based FCS_CKM.2 Selection	NDcPP21	FCS_CKM.2
TD0401	NIT Technical Decision for Reliance on external servers to meet SFRs	NDcPP21	FTP_ITC.1
TD0400	NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment	NDcPP21	FCS_CKM.1, FCS_CKM.2
TD0399	NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2)	NDcPP21	FIA_X509_EXT.2
TD0398	NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR	NDcPP21	FCS_SSHS_EXT.1.1
TD0397	NIT Technical Decision for Fixing AES-CTR Mode Tests	NDcPP21	FCS_COP.1/ DataEncryption
TD0396	NIT Technical Decision for FCS_TLSC_EXT.1.1, Test 2	NDcPP21	FCS_TLSC_EXT.2.1
TD0395	*NIT Technical Decision for Different Handling of TLS1.1 and TLS1.2*	*NDcPP21*	*Not applied because this ST does not include FCS_TLSS_EXT.2*

2.1 Conformance Rationale

The ST conforms to the NDcPP21/VPNGW10/STFFW13. As explained previously, the security problem definition, security objectives, and security requirements have been drawn from the PP.

3. Security Objectives

The Security Problem Definition may be found in the NDcPP21/VPNGW10/STFFW13 and this section reproduces only the corresponding Security Objectives for operational environment for reader convenience. The NDcPP21/VPNGW10/STFFW13 offers additional information about the identified security objectives, but that has not been reproduced here and the NDcPP21/VPNGW10/STFFW13 should be consulted if there is interest in that material.

In general, the NDcPP21/VPNGW10/STFFW13 has defined Security Objectives appropriate for firewall/VPN and as such are applicable to the PacStar 451/453/455 TOE.

3.1 Security Objectives for the Operational Environment

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.CONNECTIONS See TD0520 for SARs.
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.

OE.NO_GENERAL_PURPOSE There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

OE.NO_THRU_TRAFFIC_PROTECTION The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.

OE.PHYSICAL Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

OE.RESIDUAL_INFORMATION The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

OE.TRUSTED_ADMIN TOE Administrators are trusted to follow and apply all guidance documentation in a trusted manner.
For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.

OE.UPDATES The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

4. Extended Components Definition

All of the extended requirements in this ST have been drawn from the NDcPP21/VPNGW10/STFFW13. The NDcPP21/VPNGW10/STFFW13 defines the following extended requirements and since they are not redefined in this ST the NDcPP21/VPNGW10/STFFW13 should be consulted for more information in regard to those CC extensions.

Extended SFRs:

- NDcPP21:FAU_STG_EXT.1: Protected Audit Event Storage
- NDcPP21:FCS_HTTPS_EXT.1: HTTPS Protocol
- NDcPP21:FCS_IPSEC_EXT.1: IPsec Protocol
- VPNGW10:FCS_IPSEC_EXT.1: Internet Protocol Security (IPsec) Communications
- NDcPP21:FCS_NTP_EXT.1: NTP Protocol
- NDcPP21:FCS_RBG_EXT.1: Random Bit Generation
- NDcPP21:FCS_SSHS_EXT.1: SSH Server Protocol
- NDcPP21:FCS_TLSC_EXT.2: TLS Client Protocol with authentication
- NDcPP21:FCS_TLSS_EXT.1: TLS Server Protocol
- STFFW13:FFW_RUL_EXT.1: Stateful Traffic Filtering
- STFFW13:FFW_RUL_EXT.2: Stateful Filtering of Dynamic Protocols
- NDcPP21:FIA_PMG_EXT.1: Password Management
- VPNGW10:FIA_PSK_EXT.1: Pre-Shared Key Composition
- NDcPP21:FIA_UAU_EXT.2: Password-based Authentication Mechanism
- NDcPP21:FIA_UIA_EXT.1: User Identification and Authentication
- NDcPP21:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
- NDcPP21:FIA_X509_EXT.2: X.509 Certificate Authentication
- VPNGW10:FIA_X509_EXT.2: X.509 Certificate Authentication
- NDcPP21:FIA_X509_EXT.3: X.509 Certificate Requests
- VPNGW10:FIA_X509_EXT.3: X.509 Certificate Requests
- VPNGW10:FPF_RUL_EXT.1: Rules for Packet Filtering
- NDcPP21:FPT_APW_EXT.1: Protection of Administrator Passwords
- NDcPP21:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
- NDcPP21:FPT_STM_EXT.1: Reliable Time Stamps
- NDcPP21:FPT_TST_EXT.1: TSF testing
- VPNGW10:FPT_TST_EXT.1: TSF Testing
- VPNGW10:FPT_TST_EXT.3: TSF Self-Test with Defined Methods
- NDcPP21:FPT_TUD_EXT.1: Trusted update
- VPNGW10:FPT_TUD_EXT.1: Trusted Update
- NDcPP21:FTA_SSL_EXT.1: TSF-initiated Session Locking
- VPNGW10:FTA_VCM_EXT.1: VPN Client Management

5. Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the NDcPP21/VPNGW10/STFFW13. The refinements and operations already performed in the NDcPP21/VPNGW10/STFFW13 are not identified (e.g., highlighted) here, rather the requirements have been copied from the NDcPP21/VPNGW10/STFFW13 and any residual operations have been completed herein. Of particular note, the NDcPP21/VPNGW10/STFFW13 made a number of refinements and completed some of the SFR operations defined in the Common Criteria (CC) and that PP should be consulted to identify those changes if necessary.

The SARs are also drawn from the NDcPP21/VPNGW10/STFFW13 which includes all the SARs for EAL 1. However, the SARs are effectively refined since requirement-specific 'Assurance Activities' are defined in the NDcPP21/VPNGW10/STFFW13 that serve to ensure corresponding evaluations will yield more practical and consistent assurance than the EAL 1 assurance requirements alone. The NDcPP21/VPNGW10/STFFW13 should be consulted for the assurance activity definitions.

5.1 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by PacStar 451/453/455 TOE.

Requirement Class	Requirement Component
FAU: Security audit	NDcPP21:FAU_GEN.1: Audit Data Generation
	STFFW13:FAU_GEN.1: Security Audit Data Generation
	VPNGW10:FAU_GEN.1: Audit Data Generation
	NDcPP21:FAU_GEN.2: User identity association
	NDcPP21:FAU_STG_EXT.1: Protected Audit Event Storage
FCS: Cryptographic support	NDcPP21:FCS_CKM.1: Cryptographic Key Generation
	VPNGW10:FCS_CKM.1/IKE: Cryptographic Key Generation (for IKE Peer Authentication)
	NDcPP21:FCS_CKM.2: Cryptographic Key Establishment
	NDcPP21:FCS_CKM.4: Cryptographic Key Destruction
	NDcPP21:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption)
	VPNGW10:FCS_COP.1/DataEncryption: Cryptographic Operation (AES Data Encryption/Decryption)
	NDcPP21:FCS_COP.1/Hash: Cryptographic Operation (Hash Algorithm)
	NDcPP21:FCS_COP.1/KeyedHash: Cryptographic Operation (Keyed Hash Algorithm)
	NDcPP21:FCS_COP.1/SigGen: Cryptographic Operation (Signature Generation and Verification)
	NDcPP21:FCS_HTTPS_EXT.1: HTTPS Protocol
	NDcPP21:FCS_IPSEC_EXT.1: IPsec Protocol
	VPNGW10:FCS_IPSEC_EXT.1: Internet Protocol Security (IPsec) Communications
	NDcPP21:FCS_NTP_EXT.1: NTP Protocol
	NDcPP21:FCS_RBG_EXT.1: Random Bit Generation
	NDcPP21:FCS_SSHS_EXT.1: SSH Server Protocol
	NDcPP21:FCS_TLSC_EXT.2: TLS Client Protocol with authentication
	NDcPP21:FCS_TLSS_EXT.1: TLS Server Protocol
FDP: User data protection	STFFW13:FDP_RIP.2: Full Residual Information Protection

FFW: Firewall	STFFW13:FFW_RUL_EXT.1: Stateful Traffic Filtering
	STFFW13:FFW_RUL_EXT.2: Stateful Filtering of Dynamic Protocols
FIA: Identification and authentication	NDcPP21:FIA_AFL.1: Authentication Failure Management
	NDcPP21:FIA_PMG_EXT.1: Password Management
	VPNGW10:FIA_PSK_EXT.1: Pre-Shared Key Composition
	NDcPP21:FIA_UAU.7: Protected Authentication Feedback
	NDcPP21:FIA_UAU_EXT.2: Password-based Authentication Mechanism
	NDcPP21:FIA_UIA_EXT.1: User Identification and Authentication
	NDcPP21:FIA_X509_EXT.1/Rev: X.509 Certificate Validation
	NDcPP21:FIA_X509_EXT.2: X.509 Certificate Authentication
	VPNGW10:FIA_X509_EXT.2: X.509 Certificate Authentication
	NDcPP21:FIA_X509_EXT.3: X.509 Certificate Requests
	VPNGW10:FIA_X509_EXT.3: X.509 Certificate Requests
FMT: Security management	NDcPP21:FMT_MOF.1/ManualUpdate: Management of security functions behaviour
	NDcPP21:FMT_MOF.1/Services: Management of security functions behaviour
	NDcPP21:FMT_MTD.1/CoreData: Management of TSF Data
	NDcPP21:FMT_MTD.1/CryptoKeys: Management of TSF data
	VPNGW10:FMT_MTD.1/CryptoKeys: Management of TSF Data
	NDcPP21:FMT_SMF.1: Specification of Management Functions
	VPNGW10:FMT_SMF.1: Specification of Management Functions
	STFFW13:FMT_SMF.1/FFW: Specification of Management Functions
	NDcPP21:FMT_SMR.2: Restrictions on Security Roles
FPF: Packet Filtering	VPNGW10:FPF_RUL_EXT.1: Rules for Packet Filtering
FPT: Protection of the TSF	NDcPP21:FPT_APW_EXT.1: Protection of Administrator Passwords
	VPNGW10:FPT_FLS.1/SelfTest: Fail Secure (Self-Test Failures)
	NDcPP21:FPT_SKP_EXT.1: Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
	NDcPP21:FPT_STM_EXT.1: Reliable Time Stamps
	NDcPP21:FPT_TST_EXT.1: TSF testing
	VPNGW10:FPT_TST_EXT.1: TSF Testing
	VPNGW10:FPT_TST_EXT.3: TSF Self-Test with Defined Methods
	NDcPP21:FPT_TUD_EXT.1: Trusted update
	VPNGW10:FPT_TUD_EXT.1: Trusted Update
FTA: TOE access	NDcPP21:FTA_SSL.3: TSF-initiated Termination
	VPNGW10:FTA_SSL.3/VPN: TSF-Initiated Termination (VPN Headend)
	NDcPP21:FTA_SSL.4: User-initiated Termination
	NDcPP21:FTA_SSL_EXT.1: TSF-initiated Session Locking
	NDcPP21:FTA_TAB.1: Default TOE Access Banners
	VPNGW10:FTA_TSE.1: TOE Session Establishment
	VPNGW10:FTA_VCM_EXT.1: VPN Client Management
FTP: Trusted path/channels	NDcPP21:FTP_ITC.1: Inter-TSF trusted channel
	VPNGW10:FTP_ITC.1/VPN: Inter-TSF Trusted Channel (VPN Communications)
	NDcPP21:FTP_TRP.1/Admin: Trusted Path

Table 2 TOE Security Functional Components

5.1.1 Security audit (FAU)

5.1.1.1 Audit Data Generation (NDcPP21:FAU_GEN.1)

NDcPP21:FAU_GEN.1.1

The TSF shall be able to generate an audit record of the following auditable events:
a) Start-up and shut-down 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 2.

Requirement	Auditable Events	Additional Content
NDcPP21:FAU_GEN.1		
STFFW13:FAU_GEN.1		
VPNGW10:FAU_GEN.1		
NDcPP21:FAU_GEN.2		
NDcPP21:FAU_STG_EXT.1		
NDcPP21:FCS_CKM.1		
VPNGW10:FCS_CKM.1/IKE		
NDcPP21:FCS_CKM.2		
NDcPP21:FCS_CKM.4		
NDcPP21:FCS_COP.1/DataEncryption		
VPNGW10:FCS_COP.1/DataEncryption		
NDcPP21:FCS_COP.1/Hash		
NDcPP21:FCS_COP.1/KeyedHash		
NDcPP21:FCS_COP.1/SigGen		
NDcPP21:FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure.
NDcPP21:FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure.
VPNGW10:FCS_IPSEC_EXT.1	Session Establishment with peer	Entire packet contents of packets transmitted/received during session establishment
NDcPP21:FCS_NTP_EXT.1		
NDcPP21:FCS_RBG_EXT.1		
NDcPP21:FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.
NDcPP21:FCS_TLSC_EXT.2	Failure to establish a TLS Session.	Reason for failure.
NDcPP21:FCS_TLSS_EXT.1	Failure to establish a TLS Session.	Reason for failure.
STFFW13:FDP_RIP.2	None	

STFFW13:FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface
STFFW13:FFW_RUL_EXT.2	Dynamical definition of rule Establishment of a session	None
NDcPP21:FIA_AFL.1	Unsuccessful login attempt limit is met or exceeded.	Origin of the attempt (e.g., IP address).
NDcPP21:FIA_PMG_EXT.1		
VPNGW10:FIA_PSK_EXT.1		
NDcPP21:FIA_UAU.7		
NDcPP21:FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP21:FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
NDcPP21:FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate. Any addition, replacement or removal of trust anchors in the TOE's trust store	Reason for failure of certificate validation Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
NDcPP21:FIA_X509_EXT.2		
VPNGW10:FIA_X509_EXT.2		
NDcPP21:FIA_X509_EXT.3		
VPNGW10:FIA_X509_EXT.3		
NDcPP21:FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	
NDcPP21:FMT_MOF.1/Services		
NDcPP21:FMT_MTD.1/CoreData		
NDcPP21:FMT_MTD.1/CryptoKeys	Management of cryptographic keys.	
VPNGW10:FMT_MTD.1/CryptoKeys		
NDcPP21:FMT_SMF.1	All management activities of TSF data.	
VPNGW10:FMT_SMF.1		
STFFW13:FMT_SMF.1/FFW	All management activities of TSF data (including creation, modification and deletion of firewallrules).	None
NDcPP21:FMT_SMR.2		
VPNGW10:FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol
NDcPP21:FPT_APW_EXT.1		
VPNGW10:FPT_FLS.1/SelfTest		
NDcPP21:FPT_SKP_EXT.1		
NDcPP21: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).

NDcPP21:FPT_TST_EXT.1		
VPNGW10:FPT_TST_EXT.1		
VPNGW10:FPT_TST_EXT.3		
NDcPP21:FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	
VPNGW10:FPT_TUD_EXT.1		
NDcPP21:FTA_SSL.3	The termination of a remote session by the session locking mechanism.	
VPNGW10:FTA_SSL.3/VPN		
NDcPP21:FTA_SSL.4	The termination of an interactive session.	
NDcPP21:FTA_SSL_EXT.1	(if 'lock the session' is selected) Any attempts at unlocking of an interactive session. (if 'terminate the session' is selected) The termination of a local session by the session locking mechanism.	
NDcPP21:FTA_TAB.1		
VPNGW10:FTA_TSE.1		
VPNGW10:FTA_VCM_EXT.1		
NDcPP21: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.
VPNGW10:FTP_ITC.1/VPN		
NDcPP21:FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	

NDcPP21:FAU_GEN.1.2

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

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

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

5.1.1.2 Security Audit Data Generation (STFFW13:FAU_GEN.1)

STFFW13:FAU_GEN.1.1

See NDcPP21:FAU_GEN.1 for related events

5.1.1.3 Audit Data Generation (VPNGW10:FAU_GEN.1)

VPNGW10:FAU_GEN.1.1

See NDcPP21:FAU_GEN.1 for related events

5.1.1.4 User identity association (NDcPP21:FAU_GEN.2)

NDcPP21: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.

5.1.1.5 Protected Audit Event Storage (NDcPP21:FAU_STG_EXT.1)

NDcPP21: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.

NDcPP21:FAU_STG_EXT.1.2

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

[***TOE shall consist of a single standalone component that stores audit data locally,***]

NDcPP21: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.

5.1.2 Cryptographic support (FCS)

5.1.2.1 Cryptographic Key Generation (NDcPP21:FCS_CKM.1)

NDcPP21: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 Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3].

5.1.2.2 Cryptographic Key Generation (for IKE Peer Authentication) (VPNGW10:FCS_CKM.1/IKE)

VPNGW10:FCS_CKM.1.1/IKE

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

- FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.3 for RSA schemes,

- FIPS PUB 186-4, 'Digital Signature Standard (DSS)', Appendix B.4 for ECDSA schemes and implementing 'NIST curves' P-256, P-384 and [P521]] and [***no other key generation algorithms***] and specified cryptographic key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

5.1.2.3 Cryptographic Key Establishment (NDcPP21:FCS_CKM.2)

NDcPP21:FCS_CKM.2.1

The TSF shall perform cryptographic key establishment in accordance with a specified cryptographic key establishment method: [

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, '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, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography",

- Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3]. (TD0402 applied)

5.1.2.4 Cryptographic Key Destruction (NDcPP21:FCS_CKM.4)

NDcPP21:FCS_CKM.4.1

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

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

that meets the following: No Standard.

5.1.2.5 Cryptographic Operation (AES Data Encryption/Decryption) (NDcPP21:FCS_COP.1/DataEncryption)

NDcPP21:FCS_COP.1.1/DataEncryption

The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in [***CBC, GCM***] mode and cryptographic key sizes [***128 bits, 256 bits***] that meet the following: AES as specified in ISO 18033-3, [***CBC as specified in ISO 10116, GCM as specified in ISO 19772***].

5.1.2.6 Cryptographic Operation (AES Data Encryption/Decryption) (VPNGW10:FCS_COP.1/DataEncryption)

VPNGW10: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***].

5.1.2.7 Cryptographic Operation (Hash Algorithm) (NDcPP21:FCS_COP.1/Hash)

NDcPP21: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 message digest sizes [***160, 256, 384, 512***] that meet the following: ISO/IEC 10118-3:2004.

5.1.2.8 Cryptographic Operation (Keyed Hash Algorithm) (NDcPP21:FCS_COP.1/KeyedHash)

NDcPP21: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'.

5.1.2.9 Cryptographic Operation (Signature Generation and Verification) (NDcPP21:FCS_COP.1/SigGen)

NDcPP21: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 and 3072 bits],
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [256, 384, and 521 bits]]

that meet the following:

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

5.1.2.10 HTTPS Protocol (NDcPP21:FCS_HTTPS_EXT.1)

NDcPP21:FCS_HTTPS_EXT.1.1
The TSF shall implement the HTTPS protocol that complies with RFC 2818.

NDcPP21:FCS_HTTPS_EXT.1.2
The TSF shall implement HTTPS using TLS.

NDcPP21: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.

5.1.2.11 IPsec Protocol (NDcPP21:FCS_IPSEC_EXT.1)

NDcPP21:FCS_IPSEC_EXT.1.1
The TSF shall implement the IPsec architecture as specified in RFC 4301.

NDcPP21: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.

NDcPP21:FCS_IPSEC_EXT.1.3
The TSF shall implement [***transport mode, tunnel mode***].

NDcPP21:FCS_IPSEC_EXT.1.4
The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [***AES-CBC-128, AES-CBC-256 (specified in RFC 3602)***] together with a Secure Hash Algorithm (SHA)-based HMAC [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512***] and [***AES-GCM-128, AES-GCM-256 (specified in RFC 4106)***].

NDcPP21: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]***].

NDcPP21: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)***].

NDcPP21:FCS_IPSEC_EXT.1.7
The TSF shall ensure that [***- IKEv2 SA lifetimes can be configured by a Security Administrator based on [o length of time, where the time values can be configured within [120 to 2,147,483,647 seconds. The default is 86,400 seconds or 24] hours]***].

NDcPP21:FCS_IPSEC_EXT.1.8
The TSF shall ensure that [***- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [o number of bytes,***
o length of time, where the time values can be configured within [120 to 2,147,483,647 seconds including 28,800 seconds which is 8] hours]].

NDcPP21: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.

NDcPP21: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***].

NDcPP21:FCS_IPSEC_EXT.1.11
The TSF shall ensure that all IKE protocols implement DH Group(s) [***14 (2048-bit MODP), 19 (256-bit Random ECP), 20 (384-bit Random ECP), 24 (2048-bit MODP with 256-bit POS)***].

NDcPP21: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.

NDcPP21: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***].

NDcPP21: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: [***SAN: Fully Qualified Domain Name (FQDN), Distinguished Name (DN)***] and [***no other reference identifier type***].

5.1.2.12 Internet Protocol Security (IPsec) Communications (VPNGW10:FCS_IPSEC_EXT.1)

VPNGW10:FCS_IPSEC_EXT.1.1
No change from base PP

VPNGW10:FCS_IPSEC_EXT.1.2
No change from base PP

VPNGW10:FCS_IPSEC_EXT.1.3
The TSF shall implement [***transport mode***].

VPNGW10:FCS_IPSEC_EXT.1.4
The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [***AES-CBC-128, AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES-GCM-256 (specified in RFC 4106)***] and [***no other algorithm***] together with a Secure Hash Algorithm (SHA)-based HMAC [***HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512***].

VPNGW10:FCS_IPSEC_EXT.1.5
No change from base PP

VPNGW10:FCS_IPSEC_EXT.1.6
No change from base PP

VPNGW10:FCS_IPSEC_EXT.1.7
No change from base PP

VPNGW10:FCS_IPSEC_EXT.1.8
No change from base PP

VPNGW10:FCS_IPSEC_EXT.1.9
No change from base PP

VPNGW10:FCS_IPSEC_EXT.1.10
No change from base PP

VPNGW10:FCS_IPSEC_EXT.1.11
The TSF shall ensure that IKE protocols implement DH Groups 19 (256-bit Random ECP), 20 (384-bit Random ECP, and [***14 (2048-bit MODP), 24 (2048-bit MODP with 256-bit POS)***].

VPNGW10:FCS_IPSEC_EXT.1.12
No change from base PP

VPNGW10:FCS_IPSEC_EXT.1.13
No change from base PP

VPNGW10: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)***].

5.1.2.13 NTP Protocol (NDcPP21:FCS_NTP_EXT.1)

NDcPP21:FCS_NTP_EXT.1.1
The TSF shall use only the following NTP version(s) [***NTP v3 (RFC 1305)***].

NDcPP21:FCS_NTP_EXT.1.2
The TSF shall update its system time using [***IPsec***] to provide trusted communication between itself and an NTP time source.

NDcPP21:FCS_NTP_EXT.1.3
The TSF shall not update NTP timestamp from broadcast and/or multicast addresses

NDcPP21:FCS_NTP_EXT.1.4
The TSF shall support configuration of at least three (3) NTP time sources.

5.1.2.14 Random Bit Generation (NDcPP21:FCS_RBG_EXT.1)

NDcPP21:FCS_RBG_EXT.1.1
The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [***CTR_DRBG (AES)***].

NDcPP21:FCS_RBG_EXT.1.2
The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[one] hardware-based noise source***] with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011Table C.1 'Security Strength Table for Hash Functions', of the keys and hashes that it will generate.

5.1.2.15 SSH Server Protocol (NDcPP21:FCS_SSHS_EXT.1)

NDcPP21:FCS_SSHS_EXT.1.1
The TSF shall implement the SSH protocol that complies with RFC(s) [***4251, 4252, 4253, 4254***]. (TD0398 applied)

NDcPP21:FCS_SSHS_EXT.1.2
The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [***password-based***].

NDcPP21: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.

NDcPP21:FCS_SSHS_EXT.1.4
The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [***aes128-cbc, aes256-cbc***].

NDcPP21: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. (TD0424 applied)

NDcPP21:FCS_SSHS_EXT.1.6
The TSF shall ensure that the SSH transport implementation uses [***hmac-sha1, hmac-sha2-256***] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

NDcPP21: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.

NDcPP21: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. (TD0475 applied)

5.1.2.16 TLS Client Protocol with authentication (NDcPP21:FCS_TLSC_EXT.2)

NDcPP21:FCS_TLSC_EXT.2.1
The TSF shall implement [***TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [***TLS_DHE_RSA_WITH_AES_256_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)].

NDcPP21:FCS_TLSC_EXT.2.2
The TSF shall verify that the presented identifiers of the following types [***identifiers defined in RFC 6125***] are matched to reference identifiers. (TD0481 applied)

NDcPP21:FCS_TLSC_EXT.2.3
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***].

NDcPP21:FCS_TLSC_EXT.2.4
The TSF shall [***present the Supported Elliptic Curves Extension with the following NIST curves: [secp256r1, secp384r1, secp521r1] and no other curves***] in the Client Hello.

NDcPP21:FCS_TLSC_EXT.2.5
The TSF shall support mutual authentication using X.509v3 certificates.

5.1.2.17 TLS Server Protocol (NDcPP21:FCS_TLSS_EXT.1)

NDcPP21:FCS_TLSS_EXT.1.1
The TSF shall implement [***TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [***TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,***
TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,
TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 as defined in RFC 5246,
TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246,
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289].

NDcPP21:FCS_TLSS_EXT.1.2
The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0, and [***none***].

NDcPP21:FCS_TLSS_EXT.1.3
The TSF shall [***generate EC Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves, generate Diffie-Hellman parameters of size [2048 bits]***].

5.1.3 User data protection (FDP)

5.1.3.1 Full Residual Information Protection (STFFW13:FDP_RIP.2)

STFFW13:FDP_RIP.2.1
The TSF shall ensure that any previous information content of a resource is made unavailable upon the [***allocation of the resource to***] all objects.

5.1.4 Firewall (FFW)

5.1.4.1 Stateful Traffic Filtering (STFFW13:FFW_RUL_EXT.1)

STFFW13:FFW_RUL_EXT.1.1
The TSF shall perform stateful traffic filtering on network packets processed by the TOE.

STFFW13:FFW_RUL_EXT.1.2
The TSF shall allow the definition of stateful traffic filtering rules using the following network protocol fields:
- ICMPv4
  - o Type
  - o Code
- ICMPv6
  - o Type
  - o Code
- IPv4
  - o Source address
  - o Destination Address
  - o Transport Layer Protocol
- IPv6
  - o Source address
  - o Destination Address
  - o Transport Layer Protocol
  - o [***no other field***]
- TCP
  - o Source Port
  - o Destination Port
- UDP
  - o Source Port
  - o Destination Port

and distinct interface.

STFFW13:FFW_RUL_EXT.1.3
The TSF shall allow the following operations to be associated with stateful traffic filtering rules: permit or drop with the capability to log the operation.

STFFW13:FFW_RUL_EXT.1.4
The TSF shall allow the stateful traffic filtering rules to be assigned to each distinct network interface.

STFFW13:FFW_RUL_EXT.1.5
The TSF shall:
a) accept a network packet without further processing of stateful traffic filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [***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***].

STFFW13:FFW_RUL_EXT.1.6
The TSF shall enforce the following default stateful traffic filtering rules on all network traffic:
a) The TSF shall drop and be capable of [***logging***] packets which are invalid fragments;
b) The TSF shall drop and be capable of [***logging***] fragmented packets which cannot be re-assembled completely;
c) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a broadcast network;

d) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a multicast network;
e) The TSF shall drop and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
f) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address 'reserved for future use' (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;
g) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as an 'unspecified address' or an address 'reserved for future definition and use' (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;
h) The TSF shall drop and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and [
[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:***
  - ***a. In routed mode when the TOE receives a through-the-box:***
    - ***i. IPv4 packet with destination IP address equal to 0.0.0.0***
    - ***ii. IPv4 packet with source IP address equal to 0.0.0.0***
  - ***b. In routed or transparent mode when the TOE receives a through-the-box IPv4 packet with any of:***
    - ***i. first octet of the source IP address equal to zero***
    - ***ii. network part of the source IP address equal to all 0's***
    - ***iii. network part of the source IP address equal to all 1's***
    - ***iv. 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)]]***].

STFFW13:FFW_RUL_EXT.1.7

The TSF shall be capable of dropping and logging according to the following rules: a) The TSF shall drop and be capable of logging network packets where the source address of the network packet is equal to the address of the network interface where the network packet was received; b) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is a link-local address; c) The TSF shall drop and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received.

STFFW13:FFW_RUL_EXT.1.8

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

STFFW13:FFW_RUL_EXT.1.9

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

STFFW13:FFW_RUL_EXT.1.10

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

5.1.4.2 Stateful Filtering of Dynamic Protocols (STFFW13:FFW_RUL_EXT.2)

STFFW13:FFW_RUL_EXT.2.1

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

5.1.5 Identification and authentication (FIA)

5.1.5.1 Authentication Failure Management (NDcPP21:FIA_AFL.1)

NDcPP21: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. (TD0408 applied)

NDcPP21: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 remote session using any authentication method that involves a password until [action to unlock the account] is taken by an Administrator***]. (TD0408 applied)

5.1.5.2 Password Management (NDcPP21:FIA_PMG_EXT.1)

NDcPP21: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: [***'!', '@', '#', '$', '%', '^', '&', '*', '(', ')', [â€˜ ` (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 [***8***] and [***127***] characters.

5.1.5.3 Pre-Shared Key Composition (VPNGW10:FIA_PSK_EXT.1)

VPNGW10:FIA_PSK_EXT.1.1

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

VPNGW10:FIA_PSK_EXT.1.2

The TSF shall be able to accept text-based pre-shared keys that: - Are 22 characters and [***no other lengths***] ; - composed of any combination of upper and lower case letters, numbers, and special characters (that include: '!', '@', '#', '$', '%', '^', '&', '*', '(', and ')').

VPNGW10:FIA_PSK_EXT.1.3

The TSF shall condition the text-based pre-shared keys by using [***SHA1, SHA-256, SHA-512, [SHA-384]***].

VPNGW10:FIA_PSK_EXT.1.4

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

5.1.5.4 Protected Authentication Feedback (NDcPP21:FIA_UAU.7)

NDcPP21: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.

5.1.5.5 Password-based Authentication Mechanism (NDcPP21:FIA_UAU_EXT.2)

NDcPP21:FIA_UAU_EXT.2.1

The TSF shall provide a local [***password-based, [support for RADIUS, TACACS+]***] authentication mechanism to perform local administrative user authentication. (TD0408 applied)

5.1.5.6 User Identification and Authentication (NDcPP21:FIA_UIA_EXT.1)

NDcPP21: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***].

NDcPP21:FIA_UIA_EXT.1.2

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

5.1.5.7 X.509 Certificate Validation (NDcPP21:FIA_X509_EXT.1/Rev)

NDcPP21:FIA_X509_EXT.1.1/Rev

The TSF shall validate certificates in accordance with the following rules:
- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5***]
- The TSF shall validate the extendedKeyUsage field according to the following rules:

o 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.

o 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.

o 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.

o 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.

NDcPP21: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.

5.1.5.8 X.509 Certificate Authentication (NDcPP21:FIA_X509_EXT.2)

NDcPP21:FIA_X509_EXT.2.1

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

NDcPP21: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***].

5.1.5.9 X.509 Certificate Authentication (VPNGW10:FIA_X509_EXT.2)

VPNGW10: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***].

5.1.5.10 X.509 Certificate Requests (NDcPP21:FIA_X509_EXT.3)

NDcPP21:FIA_X509_EXT.3.1
The TSF shall generate a Certification Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [***Common Name, Organization, Organizational Unit, Country***].

NDcPP21:FIA_X509_EXT.3.2
The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.1.5.11 X.509 Certificate Requests (VPNGW10:FIA_X509_EXT.3)

VPNGW10:FIA_X509_EXT.3.1
The TSF shall generate a Certificate Request as specified by RFC 2986 and be able to provide the following information in the request: public key and [***Common Name, Organization, Organizational Unit, Country***].

VPNGW10:FIA_X509_EXT.3.2
The TSF shall validate the chain of certificates from the Root CA upon receiving the CA Certificate Response.

5.1.6 Security management (FMT)

5.1.6.1 Management of security functions behaviour (NDcPP21:FMT_MOF.1/ManualUpdate)

NDcPP21:FMT_MOF.1.1/ManualUpdate
The TSF shall restrict the ability to enable the functions to perform manual update to Security Administrators.

5.1.6.2 Management of security functions behaviour (NDcPP21:FMT_MOF.1/Services)

NDcPP21:FMT_MOF.1.1/Services
The TSF shall restrict the ability to enable and disable start and stop services to Security Administrators.

5.1.6.3 Management of TSF Data (NDcPP21:FMT_MTD.1/CoreData)

NDcPP21:FMT_MTD.1.1/CoreData
The TSF shall restrict the ability to manage the TSF data to Security Administrators.

5.1.6.4 Management of TSF data (NDcPP21:FMT_MTD.1/CryptoKeys)

NDcPP21:FMT_MTD.1.1/CryptoKeys
The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

5.1.6.5 Management of TSF Data (VPNGW10:FMT_MTD.1/CryptoKeys)

VPNGW10: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.

5.1.6.6 Specification of Management Functions (NDcPP21:FMT_SMF.1)

NDcPP21:FMT_SMF.1.1

The TSF shall be capable of performing the following management functions:
- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [***digital signature***] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [***Ability to manage the cryptographic keys,***
Ability to configure the cryptographic functionality,
Ability to configure the lifetime for IPsec SAs].

5.1.6.7 Specification of Management Functions (VPNGW10:FMT_SMF.1)

VPNGW10: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 enable, disable, determine and modify the behavior of all the security functions of the TOE identified in this PP-Module;
- Ability to configure all security management functions identified in other sections of this PP-Module; [***- No other capabilities***].

5.1.6.8 Specification of Management Functions (STFFW13:FMT_SMF.1/FFW)

STFFW13:FMT_SMF.1.1/FFW

The TSF shall be capable of performing the following management functions: â€¢ Ability to configure firewall rules.

5.1.6.9 Restrictions on Security Roles (NDcPP21:FMT_SMR.2)

NDcPP21:FMT_SMR.2.1

The TSF shall maintain the roles: - Security Administrator.

NDcPP21:FMT_SMR.2.2

The TSF shall be able to associate users with roles.

NDcPP21: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.

5.1.7 Packet Filtering (FPF)

5.1.7.1 Rules for Packet Filtering (VPNGW10:FPF_RUL_EXT.1)

VPNGW10:FPF_RUL_EXT.1.1

The TSF shall perform Packet Filtering on network packets processed by the TOE.

VPNGW10:FPF_RUL_EXT.1.2

The TSF shall allow the definition of Packet Filtering rules using the following network protocols and protocol fields:

- IPv4 (RFC 791)
  - o Source address
  - o Destination Address
  - o Protocol
- IPv6 (RFC 2460)
  - o Source address
  - o Destination Address
  - o Next Header (Protocol)
- TCP (RFC 793)
  - o Source Port
  - o Destination Port
- UDP (RFC 768)
  - o Source Port
  - o Destination Port.

VPNGW10:FPF_RUL_EXT.1.3

The TSF shall allow the following operations to be associated with Packet Filtering rules: permit and drop with the capability to log the operation.

VPNGW10:FPF_RUL_EXT.1.4

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

VPNGW10:FPF_RUL_EXT.1.5

The TSF shall process the applicable Packet Filtering rules (as determined in accordance with FPF_RUL_EXT.1.4) in the following order: Administrator-defined.

VPNGW10:FPF_RUL_EXT.1.6

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

5.1.8 Protection of the TSF (FPT)

5.1.8.1 Protection of Administrator Passwords (NDcPP21:FPT_APW_EXT.1)

NDcPP21:FPT_APW_EXT.1.1

The TSF shall store administrative passwords in non-plaintext form. (TD0483 applied)

NDcPP21:FPT_APW_EXT.1.2

The TSF shall prevent the reading of plaintext administrative passwords. (TD0483 applied)

5.1.8.2 Fail Secure (Self-Test Failures) (VPNGW10:FPT_FLS.1/SelfTest)

VPNGW10:FPT_FLS.1.1/SelfTest

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

5.1.8.3 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) (NDcPP21:FPT_SKP_EXT.1)

NDcPP21:FPT_SKP_EXT.1.1

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

5.1.8.4 Reliable Time Stamps (NDcPP21:FPT_STM_EXT.1)

NDcPP21:FPT_STM_EXT.1.1

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

NDcPP21:FPT_STM_EXT.1.2

The TSF shall [***allow the Security Administrator to set the time, synchronise time with an NTP server***].

5.1.8.5 TSF testing (NDcPP21:FPT_TST_EXT.1)

NDcPP21: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: [***FIPS 140-2 standard power-up self-tests and firmware integrity test***].

5.1.8.6 TSF Testing (VPNGW10:FPT_TST_EXT.1)

VPNGW10:FPT_TST_EXT.1.1

The TSF shall run a suite of the following self-tests [***during initial startup (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***].

5.1.8.7 TSF Self-Test with Defined Methods (VPNGW10:FPT_TST_EXT.3)

VPNGW10:FPT_TST_EXT.3.1

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

VPNGW10:FPT_TST_EXT.3.2

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

5.1.8.8 Trusted update (NDcPP21:FPT_TUD_EXT.1)

NDcPP21: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***].

NDcPP21: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***].

NDcPP21:FPT_TUD_EXT.1.3

The TSF shall provide means to authenticate firmware/software updates to the TOE using a [***digital signature mechanism***] prior to installing those updates.

5.1.8.9 Trusted Update (VPNGW10:FPT_TUD_EXT.1)

VPNGW10:FPT_TUD_EXT.1.1

VPNGW10:FPT_TUD_EXT.1.2

VPNGW10:FPT_TUD_EXT.1.3

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

5.1.9 TOE access (FTA)

5.1.9.1 TSF-initiated Termination (NDcPP21:FTA_SSL.3)

NDcPP21:FTA_SSL.3.1

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

5.1.9.2 TSF-Initiated Termination (VPN Headend) (VPNGW10:FTA_SSL.3/VPN)

VPNGW10:FTA_SSL.3.1/VPN

The TSF shall terminate a remote VPN client session after an Administrator configurable time interval of session inactivity.

5.1.9.3 User-initiated Termination (NDcPP21:FTA_SSL.4)

NDcPP21:FTA_SSL.4.1

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

5.1.9.4 TSF-initiated Session Locking (NDcPP21:FTA_SSL_EXT.1)

NDcPP21:FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions, [***terminate the session***] after a Security Administrator-specified time period of inactivity.

5.1.9.5 Default TOE Access Banners (NDcPP21:FTA_TAB.1)

NDcPP21: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.

5.1.9.6 TOE Session Establishment (VPNGW10:FTA_TSE.1)

VPNGW10:FTA_TSE.1.1

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

5.1.9.7 VPN Client Management (VPNGW10:FTA_VCM_EXT.1)

VPNGW10:FTA_VCM_EXT.1.1

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

5.1.10 Trusted path/channels (FTP)

5.1.10.1 Inter-TSF trusted channel (NDcPP21:FTP_ITC.1)

NDcPP21: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.

NDcPP21:FTP_ITC.1.2
The TSF shall permit the TSF or the authorized IT entities to initiate communication via the trusted channel.

NDcPP21: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 or TACACS+ over IPsec;
Remote VPN peer using IPsec].

5.1.10.2 Inter-TSF Trusted Channel (VPN Communications) (VPNGW10:FTP_ITC.1/VPN)

VPNGW10:FTP_ITC.1.1/VPN
The TSF shall be capable of using IPsec to provide a communication channel between itself and authorized IT entities supporting VPN communications that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

VPNGW10:FTP_ITC.1.2/VPN
The TSF shall permit the TSF or the authorized IT entities to initiate communication via the trusted channel.

VPNGW10:FTP_ITC.1.3/VPN
The TSF shall initiate communication via the trusted channel for [***Remote VPN peer]***.

5.1.10.3 Trusted Path (NDcPP21:FTP_TRP.1/Admin)

NDcPP21: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.

NDcPP21:FTP_TRP.1.2/Admin
The TSF shall permit remote Administrators to initiate communication via the trusted path.

NDcPP21:FTP_TRP.1.3/Admin
The TSF shall require the use of the trusted path for initial Administrator authentication and all remote administration actions.

5.2 TOE Security Assurance Requirements

The SARs for the TOE are the components as specified in Part 3 of the Common Criteria. Note that the SARs have effectively been refined with the assurance activities explicitly defined in association with both the SFRs and SARs.

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1: Basic Functional Specification
AGD: Guidance documents	AGD_OPE.1: Operational User Guidance
	AGD_PRE.1: Preparative Procedures
ALC: Life-cycle support	ALC_CMC.1: Labelling of the TOE
	ALC_CMS.1: TOE CM Coverage
ATE: Tests	ATE_IND.1: Independent Testing â€“ Conformance
AVA: Vulnerability assessment	AVA_VAN.1: Vulnerability Survey

Table 3 Assurance Components

5.2.1 Development (ADV)

5.2.1.1 Basic Functional Specification (ADV_FSP.1)

ADV_FSP.1.1d
The developer shall provide a functional specification.

ADV_FSP.1.2d
The developer shall provide a tracing from the functional specification to the SFRs.

ADV_FSP.1.1c
The functional specification shall describe the purpose and method of use for each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.2c
The functional specification shall identify all parameters associated with each SFR-enforcing and SFR-supporting TSFI.

ADV_FSP.1.3c
The functional specification shall provide rationale for the implicit categorization of interfaces as SFR-non-interfering.

ADV_FSP.1.4c
The tracing shall demonstrate that the SFRs trace to TSFIs in the functional specification.

ADV_FSP.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ADV_FSP.1.2e
The evaluator shall determine that the functional specification is an accurate and complete instantiation of the SFRs.

5.2.2 Guidance documents (AGD)

5.2.2.1 Operational User Guidance (AGD_OPE.1)

AGD_OPE.1.1d
The developer shall provide operational user guidance.

AGD_OPE.1.1c
The operational user guidance shall describe, for each user role, the useraccessible functions and privileges that should be controlled in a secure processing environment, including appropriate warnings.

AGD_OPE.1.2c
The operational user guidance shall describe, for each user role, how to use the available interfaces provided by the TOE in a secure manner.

AGD_OPE.1.3c
The operational user guidance shall describe, for each user role, the available functions and interfaces, in particular all security parameters under the control of the user, indicating secure values as appropriate.

AGD_OPE.1.4c
The operational user guidance shall, for each user role, clearly present each type of security-relevant event relative to the user-accessible functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.

AGD_OPE.1.5c
The operational user guidance shall identify all possible modes of operation of the TOE (including operation following failure or operational error), their consequences, and implications for maintaining secure operation.

AGD_OPE.1.6c
The operational user guidance shall, for each user role, describe the security measures to be

followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7c

The operational user guidance shall be clear and reasonable.

AGD_OPE.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.2.2 Preparative Procedures (AGD_PRE.1)

AGD_PRE.1.1d

The developer shall provide the TOE, including its preparative procedures.

AGD_PRE.1.1c

The preparative procedures shall describe all the steps necessary for secure acceptance of the delivered TOE in accordance with the developer's delivery procedures.

AGD_PRE.1.2c

The preparative procedures shall describe all the steps necessary for secure installation of the TOE and for the secure preparation of the operational environment in accordance with the security objectives for the operational environment as described in the ST.

AGD_PRE.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

AGD_PRE.1.2e

The evaluator shall apply the preparative procedures to confirm that the TOE can be prepared securely for operation.

5.2.3 Life-cycle support (ALC)

5.2.3.1 Labelling of the TOE (ALC_CMC.1)

ALC_CMC.1.1d

The developer shall provide the TOE and a reference for the TOE.

ALC_CMC.1.1c

The TOE shall be labelled with its unique reference.

ALC_CMC.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.3.2 TOE CM Coverage (ALC_CMS.1)

ALC_CMS.1.1d

The developer shall provide a configuration list for the TOE.

ALC_CMS.1.1c

The configuration list shall include the following: the TOE itself; and the evaluation evidence required by the SARs.

ALC_CMS.1.2c

The configuration list shall uniquely identify the configuration items.

ALC_CMS.1.1e

The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

5.2.4 Tests (ATE)

5.2.4.1 Independent Testing â€“ Conformance (ATE_IND.1)

ATE_IND.1.1d
The developer shall provide the TOE for testing.

ATE_IND.1.1c
The TOE shall be suitable for testing.

ATE_IND.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.

ATE_IND.1.2e
The evaluator shall test a subset of the TSF to confirm that the TSF operates as specified.

5.2.5 Vulnerability assessment (AVA)

5.2.5.1 Vulnerability Survey (AVA_VAN.1)

AVA_VAN.1.1d
The developer shall provide the TOE for testing.

AVA_VAN.1.1c
The TOE shall be suitable for testing.

AVA_VAN.1.1e
The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence..

AVA_VAN.1.2e
The evaluator shall perform a search of public domain sources to identify potential vulnerabilities in the TOE.

AVA_VAN.1.3e
The evaluator shall conduct penetration testing, based on the identified potential vulnerabilities, to determine that the TOE is resistant to attacks performed by an attacker possessing Basic attack potential.

6. TOE Summary Specification

This chapter describes the security functions:

- Security audit
- Cryptographic support
- User data protection
- Firewall
- Identification and authentication
- Security management
- Packet Filtering
- Protection of the TSF
- TOE access
- Trusted path/channels

6.1 Security audit

FAU_GEN.1

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. 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:

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.

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 Administrator Guide.

The following high-level events are auditable by the TOE:

Auditable Event	Rationale
Modifications to the group of users that are part of the authorized administrator role.	All changes to the configuration (and hence all security relevant administrator actions) are logged when the logging level is set to at least the 'notifications' level. These changes would fall into the category of configuration changes such as enabling or disabling features and services. The identity of the administrator taking the action and the user being affected (assigned

	to the authorized administrator role) are both included within the event.
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.
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.

NDcPP21: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.

NDcPP21:FAU_STG_EXT.1

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.

6.2 Cryptographic support

NDcPP21:FCS_CKM.1/VPNGW10:FCS_CKM.1/IKE, NDcPP21:FCS_CKM.2:

In the TOE cryptographic functions are used to establish TLS, HTTPS, SSHv2, and IPsec 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, "Digital Signature Standard (DSS)", Appendix B.4 and RSA with key sizes 2048 and 3072 bits according to FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3.

Key establishment for asymmetric keys on all models of the TOE implements ECDSA-based and DH-based key establishment schemes as specified in NIST SP 800-56A "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography". In addition, the TOE also supports DH group 14 key establishment scheme that meets standard RFC 3526, section 3 for interoperability.

The TOE 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 ECDSA-based and DH-based schemes. The RSA-based implementation is vendor affirmation (out of scope) and the KAS ECC and FFC.

Scheme	SFR	Services
RSA	FCS_TLSS_EXT.1, FCS_IPSEC_EXT.1, FCS_SSHS_EXT.1	HTTPS Remote Administration, SSH Remote Administration, syslog over IPsec
ECC(P-256, P-348, P-521)	FCS_TLSC_EXT.2 FCS_IPSEC_EXT.1	Syslog over TLS, Syslog over IPsec, NTP over IPsec, Connections with AAA servers
ECC (P-256, P-348, P-521)	FCS_TLSS_EXT.1	HTTPS Remote Administration
FFC	FCS_TLSC_EXT.2	Syslog over TLS
FFC	FCS_TLSS_EXT.1	HTTPS Remote Administration
Diffie-Hellman (Group 14)	FCS_SSHS_EXT.1	SSH Remote Administration

NDcPP21:FCS_CKM.4:

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 CAVP 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]

NDcPP21:FCS_COP.1/ DataEncryption, VPNGW10:FCS_COP.1/DataEncryption:

In the TOE cryptographic functions are used to establish TLS, HTTPS, SSHv2, and IPsec sessions, for IPsec traffic and authentication keys, and for IKE authentication and encryption keys.

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.

NDcPP21:FCS_COP.1/Hash, NDcPP21:FCS_COP.1/KeyedHash:

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).

NDcPP21:FCS_COP.1/SigGen:

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.

NDcPP21:FCS_RBG_EXT.1:

The ASAv virtual uses an SP 800-90 CTR_DRBG with AES-256 (Intel Secure Key) for random number generation. Random number generation in the ASAv uses hardware ring oscillators from the platform as the entropy source. More information is provided in the entropy design documentation.

NDcPP21:FCS_HTTPS_EXT.1, NDcPP21:FCS_TLSC_EXT.2, NDcPP21:FCS_TLSS_EXT.1:

The TOE implements HTTP over TLS (or HTTPS) to support remote administration using ASDM, and TLS to support secure syslog connection. A remote administrator can connect over HTTPS to the TOE with their web browser and load the ASDM software from the ASDM.

The TOE supports TLS v1.2 and TLSv1.1 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*

Follow 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

srv-id value

uri-id value

For example,

ciscoasa(config)# crypto ca reference-identity syslogServer

ciscoasa(config-ca-ref-identity)# cn-id syslog.cisco.com

To configure the syslog server certification 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

NIST curves 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 conforms to both RFCs supporting both DH 2048 bits and NIST ECC curves secp256r1, secp384r1, secp521r1.

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 implements IPsec to provide both certificates and pre-shared key-based authentication and encryption services to prevent unauthorized viewing or modification of data as it travels over the external network. The TOE implementation of the IPsec standard (in accordance with the RFCs noted in the SFR) uses the Encapsulating Security Payload (ESP) protocol to provide authentication, encryption and anti-replay services. In addition, the TOE supports both transport and tunnel modes. Transport mode is only supported for peer-to-peer IPsec connection while tunnel mode is supported for all VPN connections including remote access.

IPsec Internet Key Exchange, also called IKE, is the negotiation protocol that lets two peers agree on how to build an IPsec Security Association (SA). In the evaluated configuration, only IKEv2 is supported. The IKEv2 protocols implement Peer Authentication using the RSA, 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). Furthermore, the IKE SA lifetime limits can be configured so that no more than 200 MB of traffic can be exchanged for IKE 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 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), 24 (2048-bit MODP with 256-bit POS) 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 | 24}

asa(config-ikev2-policy)#**prf** {sha | sha256 | 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 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 "cn" to define a CN (common-name) mapping rule. Use one of the 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.

NDcPP21:FCS_NTP_EXT.1

The ASAv relies on a reliable software-based real-time clock (RTC) provided by the hypervisor. The clock's date and time can be adjusted by authorized administrators, and authorized administrators can configure the TOE to use clock updates from NTP servers. NTPv3 is supported by the TOE and the NTP timestamp is not updated from broadcast or multicast addresses. The TOE supports IPsec to secure the communication with the NTP server.

NDcPP21: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;
- Password-based authentication for administrative users;
- 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 by using the following command when enabling SSHv2 on an interface:

asa(config)#**ssh key-exchange dh-group14** {*ip_address mask* | *ipv6_address/prefix*} *interface*

6.3 User data protection

STFFW13:FDP_RIP.2:

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

6.4 Firewall

STFFW13:FFW_RUL_EXT.1:

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.

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

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);
- 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);
- 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).

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 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.

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 ASAv, however, takes into consideration the state of a packet:

- Is this a new connection?

If it is a new connection, the TOE has to check the packet against access control lists and perform other tasks to determine if the packet is allowed or denied. To perform this check, the first packet of the session goes through the "session management path," and depending on the type of traffic, it might also pass through the "control plane path." The session management path is responsible for the following tasks:

– Performing the access list checks
– Performing route lookups
– Allocating NAT translations (xlates)
– Establishing sessions in the "fast path"

The TOE creates forward and reverse flows in the fast path for TCP traffic; the TOE also creates connection state information for connectionless protocols like UDP, ICMP (when you enable ICMP inspection), 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

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 count that such packets/frames 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:
%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;

%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 address of the network packet is a multicast;

See item number 6.

9. packets where the source or destination address of the network packet is a link-local 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 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*

10. packets where the source or destination address of the network packet is defined as being an address "reserved for future use" as specified in RFC 5735 for IPv4;

%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*

11. 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** :: #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*

12. 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*

13. By default, TOE will also drop (and is capable of logging) a variety of other IP packets with invalid content including:
    - Invalid source and/or destination IP address including:
        - source or destination is the network address (e.g. 0.0.0.0)
        - source and destination address are the same (with or without the source and destination ports being the same)
        - first octet of the source IP is equal to zero
        - In the items below the "network part" and "host part" are determined by the size of the local subnet of the ingress interface (when the ASAv is in routed mode), or the subnet size of the management interface (when the ASAv is in transparent mode):
            - network part of the source IP is equal to all zeros or all ones
            - host part of the source IP is equal to all zeros or all ones
    - Invalid ICMP packets including: sequence number mismatch; invalid ICMP code, and ICMP responses unrelated to any established ICMP session

TOE administrators have control over the sequencing of access control entries (ACEs) within an access control list (ACL) to be able to set the sequence in which ACEs are applied within any ACL. The entries within an ACL are always applied in a top-down sequence, and the first entry that matches the traffic is the one that's applied, regardless of whether there may be a more precise match for the traffic further down in the ACL. By changing the ordering/numbering of entries within an ACL, the administrator chances the sequence in which the entries are compared to network traffic flows.

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".

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 ASAv 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.

STFFW13:FFW_RUL_EXT.2:

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:
    - In active mode the client initiates the control session, and the server initiates the data session to a client port provided by the client;
    - 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.
    - 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.

6.5 Identification and authentication

NDcPP21:FIA_AFL.1:

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. The recommended range the administrator should configure is between 3 and 7.

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). This applies to both password-based and public key authentication methods.

NDcPP21:FIA_PMG_EXT.1:

The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters as listed in the SFR. Minimum password length is settable by the Authorized Administrator, and support passwords of 8 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.

VPNGW10:FIA_PSK_EXT.1:

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, or HMAC-SHA-512) configured by the administrator.

NDcPP21:FIA_UIA_EXT.1:

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 through the TOE's CLI (SSH or console), 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 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.

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.

NDcPP21:FIA_UIA_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.

NDcPP21: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

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 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. 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 Administrator Guide.

6.6 Security management

NDcPP21:FMT_MOF.1/Services/ NDcPP21: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.

NDcPP21:FMT_MTD.1/CryptoKeys/VPNGW10:FMT_MTD.1/CryptoKeys:

The TOE only provides the ability for authorized administrators to access TOE data, such as audit data, configuration data, cryptographic keys, security attributes (such as cryptographic keys and certificates used in VPN), routing tables, and session thresholds.

NDcPP21: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 which has been assigned to a privilege level that is permitted to perform the relevant action.

FMT_SMF.1:

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), backup and restore of TSF data ('copy' 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 SFR are available to the Security administrators through all three administrative interfaces.

NDcPP21:FMT_SMF.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 it "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.
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:

- 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.
- 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).
- Usernames with privilege levels 2-14 can login to ASDM, and have full read-write access.
- Privilege levels cannot be customized.

- When "aaa authorization command LOCAL" has been applied to the config:
  - 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.
  - 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".

6.7 Packet Filtering

VPNGW10:FPF_RUL_EXT.1:

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.

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.

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..

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.

6.8 Protection of the TSF

NDcPP21:FPT_APW_EXT.1:

The TOE includes a Master Passphrase features that can be used to configure the TOE to encrypt all locally defined user passwords. In this manner, the TOE ensures that plaintext user passwords will not be disclosed even to administrators.

VPNGW10:FPT_FLS.1/SelfTest:

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.

NDcPP21: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. All pre-shared and symmetric keys are stored in encrypted form or are masked when showing the configuration via administrative interfaces (CLI or GUI).

NDcPP21:FPT_STM_EXT.1:

The ASAv 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.

This functionality can be set at the CLI using the 'clock' commands or in ASDM through the 'Configuration > Device Setup > System Time' page.

The ASAv relies on a reliable software-based real-time-clock (RTC) provided by the hypervisor. The clock's date and time can be adjusted by authorized administrators, and authorized administrators can configure the TOE to use clock updates from NTP servers.

FPT_TST_EXT:

The TOE run 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 VPNGW10:FPT_FLS.1.

NDcPP21:FPT_TUD_EXT.1:

The TOE (and other TOE components) have specific versions that can be queried by an administrator. When updates are made available by Cisco, an administrator can obtain and manually install those updates.

Digital signatures are used to verify software/firmware update files (to ensure they have not been modified from the originals distributed by Cisco) before they are used to update the applicable TOE components. The update process will fail if the digital signature verification process fails. Instructions on how to perform verification and update are provided in the Administrator Guide.

6.9 TOE access

NDcPP21:FTA_SSL.3/NDcPP21:FTA_SSL_EXT.1:

An administrator can configure maximum inactivity times for both local and remote administrative sessions. When a session is inactive (i.e., no session input) for the configured period of time the TOE will terminate the session, requiring the administrator to log in again to establish a new session when needed.

VPNGW10: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.

NDcPP21: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.

NDcPP21:FTA_TAB.1:

The TOE provides administrators with the capability to configure advisory banner or warning message(s) that will be displayed prior to completion of the logon process at the local console or via any remote connection (e.g., SSH or HTTPS).

VPNGW10:FTA_TSE.1:

The TOE allows for creation of acls that restrict VPN connectivity-based client's IP address (location). These acls allow customization of all 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.

VPNGW10:FTA_VCM_EXT.1:

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.

6.10 Trusted path/channels

FTP_ITC.1:

The TOE uses IPsec and/or TLS to protect communications between itself and remote entities for the following purposes:

- The TOE protects transmission of audit records when sending syslog message to a remote audit server by transmitting the message over IPsec and/or TLS.
- Connections to authentication servers (AAA servers) can be protected via IPsec tunnels. Connections with AAA servers (via RADIUS or 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:

The TOE uses SSHv2 or HTTPS (for ASDM) to provide the trusted path (with protection from disclosure and modification) for all remote administration sessions. Optionally, the TOE also supports tunneling the ASDM and/or SSH connections in IPsec VPN tunnels (peer-to-peer, or remote VPN client).

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**".

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.

Table 4: Syntax Description

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.

Keyword	**Connection Type Displayed**
up	Connections in the up state.
conn_inbound	Inbound connections.
ctiqbe	CTIQBE connections
data_in	Inbound data connections.
data_out	Outbound data connections.
finin	FIN inbound connections.
finout	FIN outbound connections.
h225	H.225 connections
h323	H.323 connections
http_get	HTTP get connections.

mgcp	MGCP connections.
nojava	Connections that deny access to Java applets.
rpc	RPC connections.
service_module	Connections being scanned by an SSM.
sip	SIP connections.
skinny	SCCP connections.
smtp_data	SMTP mail data connections.
sqlnet_fixup_data	SQL*Net data inspection engine connections.
tcp_embryonic	TCP embryonic connections.
vpn_orphan	Orphaned VPN tunneled flows.

Table 5: Connection State Types

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.

Flag	Description
a	awaiting outside ACK to SYN
A	awaiting inside ACK to SYN
b	TCP state bypass. By default, all traffic that passes through the Cisco Adaptive Security Appliance (ASAv) 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 ASAv 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 ASAv preallocates an outside back connection with this flag set. If the inside client attempts to connect back to the server, then the ASAv 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 ASAv preallocates an inside back connection with this flag set. If the outside server attempts to connect back to the client, then the ASAv denies this connection attempt. Only the inside client can use the preallocated secondary connection.
q	SQL*Net data
r	inside acknowledged FIN
R	If TCP: outside acknowledged FIN for TCP connection If UDP: UDP RPC2 Because each row of “show conn” command output represents one connection (TCP or UDP), there will be only one R flag per row.
s	awaiting outside SYN
S	awaiting inside SYN
t	SIP transient connection For a UDP connection, the value t indicates that it will timeout after one minute.
T	SIP connection For UDP connections, the value T indicates that the connection will timeout according to the value specified using the “timeout sip” command.
U	up
V	VPN orphan
W	WAAS
X	Inspected by the service module, such as a CSC SSM.
Z	Cloud Web Security

Table 6: Connection State FlagsA 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.

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

Table 7: TCP connection directionality flags

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 8: 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[2] 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.

[2] 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
skeyid	Session Encryption Key and IKE Session Authentication Key. Automatically zeroized after IKE session terminated. Storage: DRAM Overwritten with: 0x00
skeyid_d	Session Encryption Key and IKE Session Authentication Key. Automatically zeroized after IKE session terminated. Storage: DRAM Overwritten with: 0x00
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.

Critical Security Parameters (CSPs)	Zeroization Cause and Effect
	Storage: DRAM Overwritten with: 0x00
RADIUS Secret	Zeroized using the following command: **# no radius-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
All CSPs	Zeroized on-demand on all file systems via the “erase” command. Storage: NVRAM Overwritten with: 0x00

7.3 CAVP Certificate List

The TOE hardware is the PacStar 451/453/455 Series, while the software is comprised of the ASAv software image Release 9.12 The TOE includes the following Processors and the Cisco Security Crypto Virtual F6.2 (for ASAv)

Hardware Model	Processor
PacStar 451 - Xeon D Family	Intel Xeon D 1539 (Broadwell)
PacStar 451 - Xeon E Family	Intel Xeon E-2254ML (Coffee Lake)
PacStar 453 - Xeon D Family	Intel Xeon D 1559, Intel Xeon D 1539, Intel Xeon D 1557 (Broadwell)
PacStar 455 - Xeon D Family	Intel Xeon D 1559, Intel Xeon D 1539, Intel Xeon D 1557 (Broadwell)

The TOE has the following CAVP certificates:

Algorithm	**SFR**	**Cisco Security Crypto Virtual F6.2** (for ASAv)
AES CBC 128/256 GCM 128/256	FCS_COP.1/DataEncryption FCS_COP.1/DataEncryption[VPN]	A971
RSA At least 2048 bit Signature Gen & Verify Key Gen	FCS_COP.1/SigGen FCS_CKM.1 FCS_CKM.1/IKE [VPN]	A971
ECDSA curves P-256, P-384 and P-521 Signature Gen & Verify Key Gen and Verify	FCS_COP.1/SigGen FCS_CKM.1 FCS_CKM.1/IKE [VPN]	A971
FFC Scheme DSA	FCS_CKM.1	A971
Hashing SHA-1, SHA-256, SHA-384, SHA-512	FCS_COP.1/Hash	A971
Keyed Hash HMAC-SHA-1, HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512	FCS_COP.1/KeyedHash	A971
DRBG CTR_DRBG(AES)	FCS_RBG_EXT.1	A971 (CTR)
KAS ECC KAS FFC CVL	FCS_CKM.2	A971

Table 9: Algorithm Certificate Numbers