# Cisco Adaptive Security Appliances

# Security Target

Version **3.0**

**July 7, 2015**

**Americas Headquarters:**

**Cisco Systems, Inc., 170 West Tasman Drive, San Jose, CA 95134-1706 USA**

# Table of Contents

# List of Tables



# List of Figures

# List of Acronyms

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

**Table 1 Acronyms**

| Acronyms / Abbreviations | Definition |
|---|---|
| AAA | Administration, Authorization, and Accounting |
| ACL | Access Control Lists |
| AES | Advanced Encryption Standard |
| CC | Common Criteria |
| CEM | Common Evaluation Methodology |
| CM | Configuration Management |
| DHCP | Dynamic Host Configuration Protocol |
| EAL | Evaluation Assurance Level |
| EHWIC | Ethernet High-Speed WAN Interface Card |
| ESP | Encapsulating Security Payload |
| Gbps | Gigabits per second |
| GE | Gigabit Ethernet port |
| HTTPS | Hyper-Text Transport Protocol Secure |
| ICMP | Internet Control Message Protocol |
| IKE | Internet Key Exchange |
| IPsec | Internet Protocol Security |
| IT | Information Technology |
| NDPP | Network Device Protection Profile |
| OS | Operating System |
| PoE | Power over Ethernet |
| POP3 | Post Office Protocol |
| PP | Protection Profile |
| SA | Security Association |
| SFP | Small–form-factor pluggable port |
| SHA | Secure Hash Algorithm |
| SIP | Session Initiation Protocol |
| SSHv2 | Secure Shell (version 2) |
| SSM | Security Services Module |
| SSP | Security Services Processor |
| ST | Security Target |
| TCP | Transport Control Protocol |
| TOE | Target of Evaluation |
| TSC | TSF Scope of Control |
| TSF | TOE Security Function |
| TSP | TOE Security Policy |
| UDP | User Datagram Protocol |
| VLAN | Virtual Local Area Network |

| Acronyms / Abbreviations | Definition |
|---|---|
| VPN | Virtual Private Network |
| WAN | Wide Area Network |
| WIC | WAN Interface Card |

# DOCUMENT INTRODUCTION

Prepared By:

Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134

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

# 1 SECURITY TARGET INTRODUCTION

The Security Target contains the following sections:

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

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

## 1.1 ST and TOE Reference

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

**Table 2: ST and TOE Identification**

| Name | Description |
|---|---|
| ST Title | Cisco Adaptive Security Appliances |
| ST Version | 3.0 |
| Publication Date | July 7, 2015 |
| Vendor and ST Author | Cisco Systems, Inc. |
| TOE Reference | Cisco Adaptive Security Appliances |
| TOE Hardware Models | ASA 5500-X Series (5512-X, 5515-X, 5525-X, 5545-X, 5555-X),<br>ASA 5585-X Series (5585-X SSP-10, 5585-X SSP-20, 5585-X SSP-40, 5585-X SSP-60)<br>ASA Services Module (ASA-SM) on Catalyst 6500 series switches including 6503-E, 6504-E, 6509-E, and 6513-E. |
| TOE Software Version | ASA 9.4(1) and ASDM 7.4 |
| ST Evaluation Status | Completed |
| Keywords | Firewall, VPN Gateway, Router |

## 1.2 TOE Overview

The Cisco Adaptive Security Appliances TOE is a purpose-built, firewall platform with VPN capabilities. The TOE includes the hardware models as defined in Table 2 of section 1.1.

### 1.2.1 TOE Product Type

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

For firewall services, the ASA 5500-X (low to mid-range), 5585-X (high-end), and ASA-SM Series all provide application-aware stateful packet filtering firewalls. A stateful packet filtering firewall controls the flow of IP traffic by matching information contained in the headers of connection-oriented or connection-less IP packets against a set of rules specified by the

authorized administrator for firewalls. This header information includes source and destination host (IP) addresses, source and destination port numbers, and the transport service application protocol (TSAP) held within the data field of the IP packet. Depending upon the rule and the results of the match, the firewall either passes or drops the packet. The stateful firewall remembers the state of the connection from information gleaned from prior packets flowing on the connection and uses it to regulate current packets. The packet will be denied if the security policy is violated.

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

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

The TOE also provides IPsec connection capabilities. All references within this ST to “VPN” connectivity refer to the use of IPsec tunnels to secure connectivity to and/or from the TOE, for example, gateway-to-gateway[1] VPN or remote access VPN. Other uses refer to the use of IPsec connections to tunnel traffic that originates from or terminates at the TOE itself, such as for transmissions from the TOE to remote audit/syslog servers, or AAA servers, or for an additional layer of security for remote administration connections to the TOE, such as SSH or TLS connections tunneled in IPsec.

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

For management purposes, the ASDM is included. ASDM allows the ASA 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;

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

- 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.

### 1.2.2 Supported non-TOE Hardware/ Software/ Firmware

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

**Table 3: IT Environment Components**

| Component | Required | Usage/Purpose Description for TOE performance |
|---|---|---|
| Management Workstation with SSH Client | Yes | This includes any IT Environment Management workstation with a SSHv2 client installed that is used by the TOE administrator to support TOE administration through SSHv2 protected channels. Any SSHv2 client that supports SSHv2 may be used. |
| ASDM Management Platform | Yes | The ASDM 7.4 operates from any of the following operating systems:<br>• Windows Vista (x86 and x64), including Service Pack 1 and 2<br>• Windows 7 (x86 and x64)<br>• Mac OS X 10.4 - 10.6 (x86 and x64)<br>Note that that ASDM software is installed on the ASA appliance and the management platform is used to connect to the ASA and run the ASDM. The only software installed on the management platform is a Cisco ASDM Launcher. |
| 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. |
| RADIUS AAA Server | No | This includes any IT environment RADIUS AAA server that provides single-use authentication mechanisms. This can be any RADIUS AAA server that provides single-use authentication. The TOE correctly leverages the services provided by this RADIUS 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 that supports IPsec tunneling. Both VPN clients and VPN gateways can be considered to be remote tunnel endpoints. |
| NTP Server | No | The TOE supports communications with an NTP server. Connections to remote NTP servers can optionally be tunneled in IPsec. |

## 1.3 TOE DESCRIPTION

This section provides an overview of the Cisco Adaptive Security Appliances Target of Evaluation (TOE). The TOE is comprised of both software and hardware. The model is

comprised of the following: ASA 5500-X Series (5512-X, 5515-X, 5525-X, 5545-X, 5555-X), ASA 5585-X Series (5585-X SSP-10, 5585-X SSP-20, 5585-X SSP-40, 5585-X SSP-60), and ASA Services Module (ASA-SM). The software is comprised of the Adaptive Security Appliance software image Release 9.4(1), with ASDM 7.4.

The Cisco Adaptive Security Appliances that comprise the TOE have common hardware characteristics. These differing characteristics affect only non-TSF relevant functionality (such as throughput, processing speed, number and type of network connections supported, number of concurrent connections supported, and amount of storage) and therefore support security equivalency of the ASAs in terms of hardware.

**Figure 1: ASA Hardware Components**

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

- 5512-X – Two RJ-45 management Gigabit Ethernet ports, two RJ45 ports (auxiliary and console), six Gigabit Ethernet ports (expandable to twelve).
- 5515-X – Two RJ-45 management Gigabit Ethernet ports, two RJ45 ports (auxiliary and console), six Gigabit Ethernet ports (expandable to twelve).
- 5525-X – Two RJ-45 management Gigabit Ethernet ports, two RJ45 ports (auxiliary and console), six Gigabit Ethernet ports (expandable to fourteen).
- 5545-X – Two RJ-45 management Gigabit Ethernet ports, two RJ45 ports (auxiliary and console), six Gigabit Ethernet ports (expandable to fourteen).
- 5555-X – Two RJ-45 management Gigabit Ethernet ports, two RJ45 ports (auxiliary and console), six Gigabit Ethernet ports (expandable to fourteen).
- 5585-X SSP-10 – Two RJ-45 management Gigabit Ethernet ports, two RJ45 ports (auxiliary and console), eight Gigabit Ethernet ports (expandable to sixteen), two 10 Gigabit Ethernet SFP+ fiber ports (expandable to four),
- 5585-X SSP-20 – Two RJ-45 management Gigabit Ethernet ports, two RJ45 ports (auxiliary and console), eight Gigabit Ethernet ports (expandable to sixteen) and a two 10 Gigabit Ethernet SFP+ fiber ports (expandable to four)
- 5585-X SSP-40 – Two RJ-45 management Gigabit Ethernet ports, two RJ45 ports (auxiliary and console), six Gigabit Ethernet ports (expandable to twelve) and a four 10 Gigabit Ethernet SFP+ fiber ports (expandable to eight)

- 5585-X SSP-60 – Two RJ-45 management Gigabit Ethernet ports, two RJ45 ports (auxiliary and console), six Gigabit Ethernet ports (expandable to twelve) and a four 10 Gigabit Ethernet SFP+ fiber ports (expandable to eight)
- ASA-SM – Installs to Catalyst 6500 series switches including 6503-E, 6504-E, 6509-E, and 6513-E. All interfaces are logical on the ASA-SM, allowing any port on the 6500 switch to operate as a firewall port and integrating firewall security inside the network infrastructure.

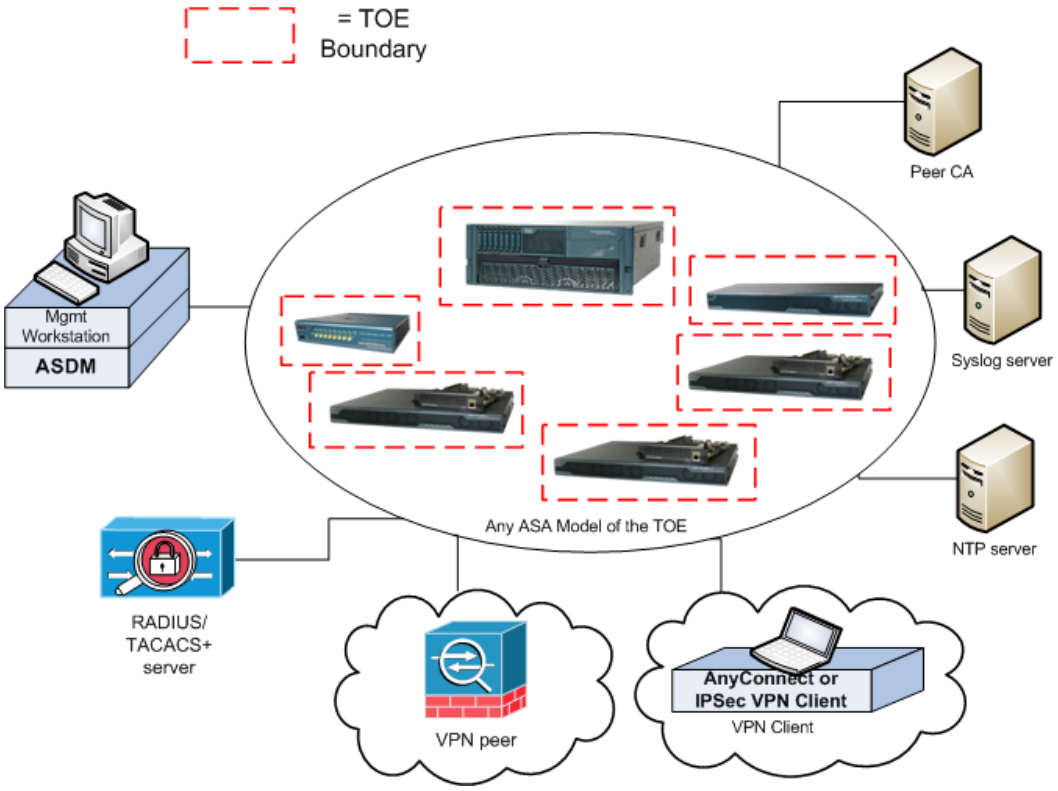
## 1.4 TOE Evaluated Configuration

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

The TOE can optionally connect to an NTP server for clock updates. 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 using its IP traffic filtering, and can encrypt traffic where necessary using TLS, SSH, and/or IPsec.

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

**Figure 2: Example TOE Deployment**



The previous figure includes the following:

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

## 1.5 Physical Scope of the TOE

The TOE is a hardware and software solution comprised of the components described in Table 4:

**Table 4 Hardware Models and Specifications**

| TOE Configuration | Hardware Configurations | Software Version |
|---|---|---|
| **ASA 5512-X**<br>**ASA 5515-X**<br>**ASA 5525-X**<br>**ASA 5545-X**<br>**ASA 5555-X**<br>  | The Cisco ASA 5500-X Adaptive Security Appliance provides high-performance firewall and VPN services and 6-14 Gigabit Ethernet interfaces, and support for up to 5,000 VPNs. | ASA release 9.4(1) |
| **ASA 5585-X SSP-10**<br>**ASA 5585-X SSP-20**<br>**ASA 5585-X SSP-40**<br>**ASA 5585-X SSP-**<br>  <br>**60** | The Cisco ASA 5585 Adaptive Security Appliance provides high-performance firewall and VPN services and 6-16 Gigabit Ethernet interfaces, 2-10 10Gigabit Ethernet interfaces, and support for up to 10,000 VPNs. | ASA release 9.4(1) |
| **ASA Services Module (ASA-SM)**<br>  | The Cisco Catalyst 6500 Series ASA Services Module supports up to: 20 Gbps maximum firewall throughput (max); 16 Gbps of maximum firewall throughput (multi-protocol); 300,000 connections per second; 10 million concurrent connections; 250 security contexts. | ASA release 9.4(1) |
| **ASDM** | Included on all ASA models with ASA | Release 7.4 |

| | | |
|---|---|---|
| | 9.4(1) | |

## 1.6 Logical Scope of the TOE

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

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

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

### 1.6.1 Security Audit

The Cisco Adaptive Security Appliances provides extensive auditing capabilities. The TOE can audit events related to cryptographic functionality, identification and authentication, and administrative actions. The Cisco Adaptive Security Appliances 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 or a configurable audit trail threshold to track the storage capacity of the audit trail. Audit logs are backed up over an encrypted channel to an external audit server.

### 1.6.2 Cryptographic Support

The TOE provides cryptography in support of other Cisco ASA 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 an entropy noise source.

### 1.6.3 Full Residual Information Protection

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

### 1.6.4 Identification and authentication

The TOE performs two types of authentication: device-level authentication of the remote device (VPN peers) and user authentication for the authorized administrator of the TOE. Device-level authentication allows the TOE to establish a secure channel with a trusted peer. The secure

channel is established only after each device authenticates the other. Device-level authentication is performed via IKE/IPsec X509v3 certificate based authentication or pre-shared key methods.

The TOE provides authentication services for administrative users wishing to connect to the TOEs secure CLI and GUI administrator interfaces. The TOE requires authorized administrators to authenticate prior to being granted access to any of the management functionality. The TOE can be configured to require a minimum password length of 15 characters as well as mandatory password complexity rules. 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 or SSH and HTTPS interfaces. The SSHv2 interface also supports authentication using SSH keys. The TOE optionally supports use of any RADIUS AAA server (part of the IT Environment) for authentication of administrative users attempting to connect to the TOE.

### 1.6.5 Security Management

The TOE provides secure administrative services for management of general TOE configuration and the security functionality provided by the TOE. All TOE administration occurs either through a secure SSHv2 or TLS/HTTPS session, or via a local console connection. The TOE provides the ability to securely manage all TOE administrative users; all identification and authentication; all audit functionality of the TOE; all TOE cryptographic functionality; the timestamps maintained by the TOE; 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 an administrative session is initially established, the TOE displays an administrator-configurable warning banner. This is used to provide any information deemed necessary by the administrator. After a configurable period of inactivity, administrative sessions will be terminated, requiring administrators to re-authenticate.

### 1.6.6 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 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 is used as the timestamp that is applied to audit records generated by the TOE. Administrators can update the TOE’s clock manually, or can configure the TOE to use NTP to synchronize the TOE’s clock with an external time source. Additionally, the TOE performs testing to verify correct operation of the appliance itself and that of the cryptographic module. Whenever any system failures occur within the TOE the TOE will cease operation.

### 1.6.7 TOE Access

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

### 1.6.8 Trusted path/Channels

The TOE supports establishing trusted paths between itself and remote administrators using SSHv2 for CLI access, and TLS/HTTPS for GUI/ASDM access. 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). The TOE can establish trusted paths of peer-to-peer VPN tunnels using IPsec, and VPN client tunnels using IPsec or TLS. Note that the VPN client is in the operational environment.

### 1.6.9 Filtering

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

The TOE also provides packet filtering and secure IPsec tunneling. The tunnels can be established between two trusted VPN peers as well as between remote VPN clients and the TOE. More accurately, these tunnels are sets of security associations (SAs). The SAs define the protocols and algorithms to be applied to sensitive packets and specify the keying material to be used. SAs are unidirectional and are established per the ESP security protocol. An authorized

administrator can define the traffic that needs to be protected via IPsec by configuring access lists (permit, deny, log) and applying these access lists to interfaces using crypto map set.

## 1.7 Excluded Functionality

The following functionality is excluded from the evaluation.

**Table 5: Excluded Functionality**

| Excluded Functionality | Exclusion Rationale |
|---|---|
| Secure Policy Manager is excluded from the evaluated configuration | Use of Security Policy Manager is beyond the scope of this Common Criteria evaluation. |
| Filtering of non-IP traffic provided by the EtherType option when configuring information flow policies is excluded from the evaluated configuration | Use of non-IP traffic filtering is beyond the scope of this Common Criteria evaluation. |
| Smart Call Home. The Smart Call Home feature provides personalized, e-mail-based and web-based notification to customers about critical events involving their individual systems. | Use of Smart Call Home is beyond the scope of this Common Criteria evaluation. |

These services will be disabled by configuration. The exclusion of this functionality does not affect compliance to the U.S. Government Protection Profiles for Security Requirements for Network Devices (NDPP), Traffic Filter Firewall Extended Package (TFFWEP), and VPN Gateway Extended Package (VPNGWEP).

# 2 CONFORMANCE CLAIMS

## 2.1 Common Criteria Conformance Claim

The TOE and ST are compliant with the Common Criteria (CC) Version 3.1, Revision 4, dated: September 2012. For a listing of Assurance Requirements claimed see section 5.6.

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

## 2.2 Protection Profile Conformance

The TOE and ST are conformant with the Protection Profiles as listed in Table 6 below:

Table 6: Protection Profiles

| Protection Profile | Version | Date |
|---|---|---|
| Security Requirements for Network Devices Errata #3 | 1.1 | 3 November 2014 |
| Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Filter Firewall | 1.0 | 19 December 2011 |
| Network Device Protection Profile (NDPP) Extended Package VPN Gateway | 1.1 | 12 April 2013 |

### 2.2.1 Protection Profile Additions

The following requirement was modified:

- FAU_GEN.1 – Additional auditable events were added. These were added in order to be compliant with the TFFWEP section 4.2.2 and VPNGWEP section 4.2.9.
- FMT_SMF.1 – Additional management functions were added such as configure firewall rules and configure VPN settings. There were added in order to be compliant with the TFFWEP section 4.2.3 and VPNGWEP section 4.2.1.5.

## 2.3 Protection Profile Conformance Claim Rationale

### 2.3.1 TOE Appropriateness

The TOE provides all of the functionality at a level of security commensurate with that identified in the U.S. Government Protection Profiles:

- U.S. Government Protection Profiles for Security Requirements for Network Devices (NDPP), Traffic Filter Firewall Extended Package (TFFWEP), and VPN Gateway Extended Package (VPNGWEP).

### 2.3.2 TOE Security Problem Definition Consistency

The Assumptions, Threats, and Organization Security Policies included in the Security Target represent the Assumptions, Threats, and Organization Security Policies specified in the U.S. Government Protection Profiles for Security Requirements for Network Devices (NDPP), Traffic Filter Firewall Extended Package (TFFWEP), and VPN Gateway Extended Package

(VPNGWEP) for which conformance is claimed verbatim. All concepts covered in the Protection Profile Security Problem Definition are included in the Security Target Statement of Security Objectives Consistency.

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

### 2.3.3 Statement of Security Requirements Consistency

The Security Functional Requirements included in the Security Target represent the Security Functional Requirements specified in the U.S. Government Protection Profiles for Security Requirements for Network Devices (NDPP), Traffic Filter Firewall Extended Package (TFFWEP), VPN Gateway Extended Package (VPNGWEP) for which conformance is claimed verbatim and several additional Security Functional Requirements are included as a result. All concepts covered the Protection Profile's Statement of Security Requirements are included in the Security Target. Additionally, the Security Assurance Requirements included in the Security Target are identical to the Security Assurance Requirements included in section 4.3 of the NDPP.

# 3 SECURITY PROBLEM DEFINITION

This chapter identifies the following:

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

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

## 3.1 Assumptions

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

**Table 7 TOE Assumptions**

| Assumption | Assumption Definition |
|---|---|
| **Reproduced from the NDPP** | |
| A.NO_GENERAL_PURPOSE | It is assumed that 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. |
| A.PHYSICAL | Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment. |
| A.TRUSTED_ADMIN | TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner. |
| **Reproduced from the TFFWEP and VPNGWEP** | |
| A.CONNECTIONS | It is assumed that the TOE is connected to distinct networks in a manner that ensures that the TOE security policies will be enforced on all applicable network traffic flowing among the attached networks. |

## 3.2 Threats

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

**Table 8 Threats**

| Threat | Threat Definition |
|---|---|
| **Reproduced from the NDPP** | |
| T.ADMIN_ERROR | An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms. |
| T.TSF_FAILURE | Security mechanisms of the TOE may fail, leading to a compromise of the TSF. |

| Threat | Threat Definition |
|---|---|
| T.UNDETECTED_ACTIONS | Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated. |
| T.UNAUTHORIZED_ACCESS | A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process, or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data. |
| T.UNAUTHORIZED_UPDATE | A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE. |
| T.USER_DATA_REUSE | User data may be inadvertently sent to a destination not intended by the original sender. |
| **Reproduced from the TFFWEP and VPNGWEP** | |
| T.NETWORK_DISCLOSURE | Sensitive information on a protected network might be disclosed resulting from ingress- or egress-based actions. |
| T. NETWORK_ACCESS | Unauthorized access may be achieved to services on a protected network from outside that network, or alternately services outside a protected network from inside the protected network. |
| T.NETWORK_MISUSE | Access to services made available by a protected network might be used counter to Operational Environment policies. |
| **Reproduced from the TFFWEP** | |
| T.NETWORK_DOS | Attacks against services inside a protected network, or indirectly by virtue of access to malicious agents from within a protected network, might lead to denial of services otherwise available within a protected network. |
| **Reproduced from the VPNGWEP** | |
| T.REPLAY_ATTACK | If malicious or external IT entities are able to gain access to the network, they may have the ability to capture information traversing throughout the network and send them on to the intended receiver. |
| T.DATA_INTEGRITY | A malicious party attempts to change the data being sent – resulting in loss of integrity. |
| T.UNAUTHORIZED_CONNECTION | While a VPN client may have the necessary credentials (e.g., certificate, pre-shared key) to connect to a VPN gateway, there may be instances where the remote client, or the machine the client is operating on, has been compromised and attempts to make unauthorized connections. |
| T.HIJACKED_SESSION | There may be an instance where a remote client's session is hijacked due to session activity. This could be accomplished because a user has walked away from the machine that was used to establish the session. |

| Threat | Threat Definition |
|---|---|
| T.UNPROTECTED_TRAFFIC | A remote machine's network traffic may be exposed to a hostile network. A user may be required to use a hostile (or unknown) network to send network traffic without being able to route the traffic appropriately. |

## 3.3 Organizational Security Policies

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

**Table 9 Organizational Security Policies**

| Policy Name | Policy Definition |
|---|---|
| **Reproduced from the NDPP** | |
| P.ACCESS_BANNER | The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE. |

# 4 SECURITY OBJECTIVES

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

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

## 4.1 Security Objectives for the TOE

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

**Table 10 Security Objectives for the TOE**

| TOE Objective | TOE Security Objective Definition |
|---|---|
| **Reproduced from the NDPP** | |
| O.PROTECTED_COMMUNICATIONS | The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities. |
| O.VERIFIABLE_UPDATES | The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the administrator to be unaltered and (optionally) from a trusted source. |
| O.SYSTEM_MONITORING | The TOE will provide the capability to generate audit data and send those data to an external IT entity. |
| O.DISPLAY_BANNER | The TOE will display an advisory warning regarding use of the TOE. |
| O.TOE_ADMINISTRATION | The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators. |
| O.RESIDUAL_INFORMATION_CLEARING | The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated. |
| O.SESSION_LOCK | The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked. |
| O.TSF_SELF_TEST | The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly. |
| **Reproduced from the TFFWEP and VPNGWEP** | |
| O.ADDRESS_FILTERING | The TOE will provide the means to filter and log network packets based on source and destination addresses. |
| O.PORT_FILTERING | The TOE will provide the means to filter and log |

| TOE Objective | TOE Security Objective Definition |
|---|---|
| | network packets based on source and destination transport layer ports. |
| **Reproduced from the TFFWEP** | |
| O.STATEFUL_INSPECTION | The TOE will determine if a network packet belongs to an allowed established connection before applying the ruleset. |
| O.RELATED_CONNECTION_FILTERING | For specific protocols, the TOE will dynamically permit a network packet flow in response to a connection permitted by the ruleset. |
| **Reproduced from the VPNGWEP** | |
| O.AUTHENTICATION | The TOE will provide a means to authenticate the user to ensure they are communicating with an authorized external IT entity. |
| O.CRYPTOGRAPHIC_FUNCTIONS | The TOE will provide means to encrypt and decrypt data as a means to maintain confidentiality and allow for detection and modification of TSF data that is transmitted outside of the TOE. |
| O.FAIL_SECURE | Upon a self-test failure, the TOE will shutdown to ensure data cannot be passed while not adhering to the security policies configured by the administrator. |
| O.CLIENT_ESTABLISHMENT_CONSTRAINTS | To address the concern that a remote client may be compromised and attempt to establish connections with the headend VPN gateway outside of “normal” operations, this objective specifies conditions under which a remote client may establish connections. The administrator may configure the headend VPN gateway to accept client’s request for a connection based on attributes the administrator feels are appropriate. |
| O.REMOTE_SESSION_TERMINATION | A remote client’s session can become vulnerability when there is a lack of activity. This is primarily due to a user walking away from a device that has a remote connection established. While some devices have a “lock screen” or logout capability, they cannot always assumed to be configured or available. To address this concern, a session termination capability is necessary during an administrator specified time period. |
| O.ASSIGNED_PRIVATE_ADDRESS | There are instances where a remote client desires secure communication with a gateway that is trusted. While a user may be connected via an untrusted network, it should still be possible to ensure that it can communicate with a known entity that controls the routing of the client’s network packets. This can be accomplished by |

| TOE Objective | TOE Security Objective Definition |
|---|---|
| | the VPN headend assigning an IP address that the gateway controls, as well as providing a routing point for the client's network traffic. |

## 4.2 Security Objectives for the Environment

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

**Table 11 Security Objectives for the Environment**

| Environment Security Objective | IT Environment Security Objective Definition |
|---|---|
| **Reproduced from the NDPP** | |
| 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.PHYSICAL | Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment. |
| OE.TRUSTED_ADMIN | TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner. |
| **Reproduced from the TFFWEP and VPNGWEP** | |
| OE.CONNECTIONS | TOE administrators will ensure that the TOE is installed in a manner that will allow the TOE to effectively enforce its policies on network traffic flowing among attached networks. |

# 5 SECURITY REQUIREMENTS

This section identifies the Security Functional Requirements for the TOE. The Security Functional Requirements included in this section are derived from Part 2 of the *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, dated: September 2012* and all international interpretations.

## 5.1 Conventions

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

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

Extended SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs. Formatting conventions outside of operations and iterations matches the formatting specified within the NDPP.

## 5.2 TOE Security Functional Requirements

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

**Table 12 Security Functional Requirements**

| Class Name | Component Identification | Component Name |
|---|---|---|
| **Reproduced from the NDPP** | | |
| FAU: Security Audit | FAU_GEN.1 | Audit Data Generation |
| | FAU_GEN.2 | User Identity Association |
| | FAU_STG_EXT.1 | External Audit Trail Storage |
| FCS: Cryptographic Support | FCS_CKM.1(1) | Cryptographic Key Generation (for asymmetric keys) |
| | FCS_CKM_EXT.4 | Cryptographic Key Zeroization |
| | FCS_COP.1(1) | Cryptographic Operation (for data encryption/decryption) |
| | FCS_COP.1(2) | Cryptographic Operation (for cryptographic signature) |
| | FCS_COP.1(3) | Cryptographic Operation (for cryptographic hashing) |
| | FCS_COP.1(4) | Cryptographic Operation (for keyed-hash message authentication) |
| | FCS_HTTPS_EXT.1 | Explicit: HTTPS |
| | FCS_RBG_EXT.1 | Extended: Cryptographic Operation (Random Bit Generation) |

| Class Name | Component Identification | Component Name |
|---|---|---|
| | FCS_TLS_EXT.1 | Explicit: TLS |
| | FCS_SSH_EXT.1 | Explicit: SSH |
| FDP: User Data Protection | FDP_RIP.2 | Full Residual Information Protection |
| FIA: Identification and Authentication | FIA_PMG_EXT.1 | Password Management |
| | FIA_UIA_EXT.1 | User Identification and Authentication |
| | FIA_UAU_EXT.2 | Extended: Password-based Authentication Mechanism |
| | FIA_UAU.7 | Protected Authentication Feedback |
| FMT: Security Management | FMT_MTD.1 | Management of TSF Data (for general TSF data) |
| | FMT_SMF.1 | Specification of Management Functions |
| | FMT_SMR.2 | Restrictions on Security Roles |
| FPT: Protection of the TSF | FPT_SKP_EXT.1 | Extended: Protection of TSF Data (for reading of all symmetric keys) |
| | FPT_APW_EXT.1 | Extended: Protection of Administrator Passwords |
| | FPT_STM.1 | Reliable Time Stamps |
| | FPT_TUD_EXT.1 | Extended: Trusted Update |
| | FPT_TST_EXT.1 | Extended: TSF Testing |
| | FPT_ITT.1 | Basic Internal TSF Data Transfer Protection |
| FTA: TOE Access | FTA_SSL_EXT.1 | TSF-initiated Session Locking |
| | FTA_SSL.3(1) | TSF-initiated Termination [for Adminstrators] |
| | FTA_SSL.4 | User-initiated Termination |
| | FTA_TAB.1 | Default TOE Access Banners |
| FTP: Trusted path/channels | FTP_ITC.1 | Inter-TSF Trusted Channel |
| | FTP_TRP.1 | Trusted Path |
| **Reproduced from the TFFWEP** | | |
| FFW: Stateful Traffic Filtering | FFW_RUL_EXT.1 | Stateful Traffic Filtering |
| **Reproduced from the VPNGWEP** | | |
| FCS: Cryptographic Support | FCS_CKM.1(2) | Cryptographic Key Generation (for asymmetric keys [used for IKE peer authentication]) |
| | FCS_IPSEC_EXT.1[2] | Extended: Internet Protocol Security (IPsec) Communications |
| FIA: Identification and Authentication | FIA_AFL.1 | Authentication Failure Handling |
| | FIA_PSK_EXT.1 | Extended: Pre-Shared Key Composition |
| | FIA_X509_EXT.1 | Extended: X.509 Certificates |
| FMT: Security Management | FMT_MOF.1 | Management of Security Functions Behavior |
| FPF: Packet Filtering | FPF_RUL_EXT.1 | Packet Filtering [specific to VPN tunnels, and distinct from FFW_RUL_EXT.1] |
| FPT: Protection of the | FPT_FLS.1 | Fail Secure |

[2] "The set of the IPsec requirements specified here take precedent over the IPsec requirements specified in the NDPP."

| Class Name | Component Identification | Component Name |
|---|---|---|
| TSF | | |
| FTA: TOE Access | FTA_SSL.3(2) | TSF-initiated Termination [for VPN Clients] |
| | FTA_TSE.1 | TOE Session Establishment |
| | FTA_VCM_EXT.1 | VPN Client Management |

## 5.3 SFRs Drawn from NDPP

### 5.3.1 Security audit (FAU)

#### 5.3.1.1 FAU_GEN.1 Audit data generation

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the not specified level of audit; and
*c) All administrative actions;*
*d) [Specifically defined auditable events listed in Table 13]*.

**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 PP/ST, [*information specified in column three of* Table 13].

**Table 13 Auditable Events**

| SFR | Auditable Event | Additional Audit Record Contents |
|---|---|---|
| FAU_GEN.1 | None. | |
| FAU_GEN.2 | None. | |
| FAU_STG_EXT.1 | None. | |
| FCS_CKM.1(1) | None. | |
| FCS_CKM.1(2) | None. | |
| FCS_CKM_EXT.4 | None. | |
| FCS_COP.1(1) | None. | |
| FCS_COP.1(2) | None. | |
| FCS_COP.1(3) | None. | |
| FCS_COP.1(4) | None. | |
| FCS_HTTPS_EXT.1 | Failure to establish an HTTPS session.<br>Establishment/Termination of an HTTPS session. | Reason for failure.<br>Non-TOE endpoint of connection (IP address) for both successes and failures. |
| FCS_IPSEC_EXT.1 | Failure to establish an IPsec SA.<br>Establishment/Termination of an IPsec SA. | Reason for failure.<br>Non-TOE endpoint of connection (IP address) for both successes and failures. |
| FCS_RBG_EXT.1 | None. | |
| FCS_SSH_EXT.1 | Failure to establish an SSH session | Reason for failure. |

| SFR | Auditable Event | Additional Audit Record Contents |
|---|---|---|
| | Establishment/Termination of an SSH session. | Non-TOE endpoint of connection (IP address) for both successes and failures. |
| FCS_TLS_EXT.1 | Failure to establish an TLS session<br>Establishment/Termination of an TLS session. | Reason for failure.<br>Non-TOE endpoint of connection (IP address) for both successes and failures. |
| FDP_RIP.2 | None. | |
| FIA_PMG_EXT.1 | None. | |
| FIA_UIA_EXT.1 | All use of the identification and authentication mechanism. | Provided user identity, origin of the attempt (e.g., IP address). |
| FIA_UAU_EXT.2 | All use of the authentication mechanism. | Origin of the attempt (e.g., IP address). |
| FIA_UAU.7 | None. | |
| FMT_MTD.1 | None. | |
| FMT_SMF.1 | None. | |
| FMT_SMR.2 | None. | |
| FPT_SKP_EXT.1 | None. | |
| FPT_APW_EXT.1 | None. | |
| FPT_ITT.1 | None. | None. |
| FPT_STM.1 | Changes to the time. | The old and new values for the time.<br>Origin of the attempt (e.g., IP address). |
| FPT_TUD_EXT.1 | Initiation of update. | No additional information. |
| FPT_TST_EXT.1 | Indication that TSF self-test was completed. | Any additional information generated by the tests beyond “success” or “failure”. |
| FTA_SSL_EXT.1 | Any attempts at unlocking of an interactive session. | No additional information. |
| FTA_SSL.3(1) | The termination of a remote session by the session locking mechanism. | No additional information. |
| FTA_SSL.4 | The termination of an interactive session. | No additional information. |
| FTA_TAB.1 | None. | |
| FTP_ITC.1 | Initiation of the trusted channel.<br>Termination of the trusted channel.<br>Failure of the trusted channel functions. | Identification of the initiator and target of failed trusted channels establishment attempt |
| FTP_TRP.1 | Initiation of the trusted channel.<br>Termination of the trusted channel.<br>Failures of the trusted path functions. | Identification of the claimed user identity. |
| FFW_RUL_EXT.1* | Application of rules configured with the ‘log’ operation | Source and destination addresses<br>Source and destination ports<br>Transport Layer Protocol<br>TOE Interface |
| | Indication of packets dropped due to too much network traffic | TOE interface that is unable to process packets |

| SFR | Auditable Event | Additional Audit Record Contents |
|---|---|---|
| FCS_IPSEC_EXT.1** | Session Establishment with peer | Source and destination addresses<br>Source and destination ports<br>TOE Interface |
| FIA_X509_EXT.1** | Establishing session with CA | Source and destination addresses<br>Source and destination ports<br>TOE Interface |
| FPF_RUL_EXT.1** | Application of rules configured with the 'log' operation | Source and destination addresses<br>Source and destination ports<br>Transport Layer Protocol<br>TOE Interface |
| | Indication of packets dropped due to too much network traffic | TOE interface that is unable to process packets |

* - Reproduced from the TFFWEP

** - Reproduced from the VPNGWEP

#### 5.3.1.2 FAU_GEN.2 User Identity Association

**FAU_GEN.2.1** For audit events resulting from actions of identified users, the TSF shall be able to associate each auditable event with the identity of the user that caused the event.

#### 5.3.1.3 FAU_STG_EXT.1 External Audit Trail Storage

**FAU_STG_EXT.1.1** The TSF shall be able to [transmit the generated audit data to an external IT entity] using a trusted channel implementing the [IPsec] protocol.

### 5.3.2 Cryptographic Support (FCS)

#### 5.3.2.1 FCS_CKM.1(1) Cryptographic Key Generation (for asymmetric keys)

**FCS_CKM.1.1(1) Refinement:** The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance with

> [NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes]

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

#### 5.3.2.1 FCS_CKM.1(2) Cryptographic Key Generation (for asymmetric keys)

**FCS_CKM.1.1(2) Refinement:** The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance with

- NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for elliptic curve-based key establishment schemes and implementing "NIST curves" P-256, P-384 and [no other curves] (as defined in FIPS PUB 186-3, "Digital Signature Standard")
- NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" for finite field-based key establishment schemes;
- [NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes]

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

*Application Note: This SFR was reproduced from the VPNGWEP and is only applicable to IKE/IPsec. IKE/IPsec supports both ECDSA and RSA key establishment. SSHv2 only supports RSA.*

#### 5.3.2.2 FCS_CKM_EXT.4 Cryptographic Key Zeroization

**FCS_CKM_EXT.4.1** The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

#### 5.3.2.3 FCS_COP.1(1) Cryptographic Operation (for data encryption/decryption)

**FCS_COP.1.1(1) Refinement:** The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [*AES operating in* [*CBC and GCM modes*]] and cryptographic key sizes 128-bits, 256-bits, and [192 bits] that meets the following:

- FIPS PUB 197, "Advanced Encryption Standard (AES)"
- [NIST SP 800-38A, NIST SP 800-38D]

*Application Note: The VPNGWEP requires that IKE/IPsec supports AES in CBC and GCM modes. SSHv2 only supports AES in CBC mode.*

#### 5.3.2.4 FCS_COP.1(2) Cryptographic Operation (for cryptographic signature)

FCS_COP.1.1(2) **Refinement:** The TSF shall perform **cryptographic signature services** in accordance with a [

- RSA Digital Signature Algorithm (RSA) with a key size (modulus) of 2048 bits or greater that meets FIPS PUB 186-2 or FIPS PUB 186-3, "Digital Signature Standard"

- Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater] that meets FIPS PUB 186-3, "Digital Signature Standard" with "NIST curves" P-256, P-384 and [no other curves] (as defined in FIPS PUB 186-3, "Digital Signature Standard") ]

*Application Note: This SFR was reproduced from the VPNGWEP and is applicable to the digital signature used in X509v3 certificates for IKE/IPsec. IKE/IPsec supports both ECDSA and RSA signed certificates. SSHv2 and trusted update only support RSA signed certificate.*

#### 5.3.2.5 FCS_COP.1(3) Cryptographic Operation (for cryptographic hashing)

FCS_COP.1.1(3) **Refinement:** 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] **bits** that meet the following: *FIPS Pub 180-3, "Secure Hash Standard."*

#### 5.3.2.6 FCS_COP.1(4) Cryptographic Operation (for keyed-hash message authentication)

FCS_COP.1.1(4) **Refinement:** The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm HMAC-[SHA-1, SHA-256, SHA-384, SHA-512]**, key size** [*160, 256, 512*]**, and message digest sizes** [160, 256, 384, 512] **bits** that meet the following: *FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."*

#### 5.3.2.7 FCS_HTTPS_EXT.1 Explicit: HTTPS

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

**FCS_HTTPS_EXT.1.2** The TSF shall implement HTTPS using TLS as specified in FCS_TLS_EXT.1.

#### 5.3.2.8 FCS_IPSEC_EXT.1 Extended: Internet Protocol Security (IPsec) Communications

*Application Note: The VPNGWEP's FCS_IPSEC_EXT.1 takes precedent over the NDPP.*

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

**FCS_IPSEC_EXT.1.2** The TSF shall implement [tunnel mode].

**FCS_IPSEC_EXT.1.3** The TSF shall have a nominal, final entry in the SPD that matches anything that is otherwise unmatched, and discards it.

**FCS_IPSEC_EXT.1.4** The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-GCM-128, AES-GCM-256 as specified in RFC

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

**FCS_IPSEC_EXT.1.5** The TSF shall implement the protocol: [IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23) and [no other RFCs for hash functions]].

**FCS_IPSEC_EXT.1.6** The TSF shall ensure the encrypted payload in the [IKEv2] protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and [AES-GCM-128, AES-GCM-256 as specified in RFC 5282].

**FCS_IPSEC_EXT.1.7** The TSF shall ensure that IKEv1 Phase 1 exchanges use only main mode.

**FCS_IPSEC_EXT.1.8 Refinement:** The TSF shall ensure that [IKEv2 SA lifetimes can be configured by an Administrator based on number of ~~packets~~ **kilobytes** or length of time, where the time values can be limited to: 24 hours for Phase 1 SAs and 8 hours for Phase 2 SAs].

*Application Note:IKEv2 SA Phase 1 can be limited by time only. IKEv2 SA Phase 2 can be limited by time or number of kilobytes. The time is in number of seconds.*

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

**FCS_IPSEC_EXT.1.10** The TSF shall generate nonces used in IKE exchanges in a manner such that the probability that a specific nonce value will be repeated during the life a specific IPsec SA is less than 1 in 2^[*9*].

**FCS_IPSEC_EXT.1.11** The TSF shall ensure that all IKE protocols implement DH Groups 14 (2048-bit MODP), 19 (256-bit Random ECP) and [

- 24 (2048-bit MODP with 256-bit POS),
- 20 (384-bit Random ECP)].

**FCS_IPSEC_EXT.1.12** The TSF shall ensure that all IKE protocols perform peer authentication using a [RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [Pre-shared Keys].

**FCS_IPSEC_EXT.1.13** 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.

#### 5.3.2.9 FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)

**FCS_RBG_EXT.1.1** The TSF shall perform all random bit generation (RBG) services in accordance with [NIST Special Publication 800-90 using Hash_DRBG with SHA-512] seeded by an entropy source that accumulated entropy from a TSF-hardware based noise source, and [a software-based noise source].

**FCS_RBG_EXT.1.2** The deterministic RBG shall be seeded with a minimum of [256 bits] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

#### 5.3.2.10 FCS_SSH_EXT.1 Explicit: SSH

**FCS_SSH_EXT.1.1** The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, and 4254, and [no other RFCs].

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

**FCS_SSH_EXT.1.3** The TSF shall ensure that, as described in RFC 4253, packets greater than [*65,535 bytes*] bytes in an SSH transport connection are dropped.

**FCS_SSH_EXT.1.4** The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: AES-CBC-128, AES-CBC-256, [no other algorithms].

**FCS_SSH_EXT.1.5** The TSF shall ensure that the SSH transport implementation uses [SSH_RSA] and [no other public key algorithms] as its public key algorithm(s).

**FCS_SSH_EXT.1.6** The TSF shall ensure that data integrity algorithms used in SSH transport connection is [hmac-sha1, hmac-sha1-96].

**FCS_SSH_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.

#### 5.3.2.11 FCS_TLS_EXT.1 Explicit: TLS

**FCS_TLS_EXT.1.1** The TSF shall implement one or more of the following protocols [TLS 1.0 (RFC 2246), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

Mandatory Ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA

Optional Ciphersuites:

[TLS_RSA_WITH_AES_256_CBC_SHA

TLS_DHE_RSA_WITH_AES_128_CBC_SHA

TLS_DHE_RSA_WITH_AES_256_CBC_SHA

TLS_RSA_WITH_AES_128_CBC_SHA256

TLS_RSA_WITH_AES_256_CBC_ SHA256

TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256

TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384].

### 5.3.3 User data protection (FDP)

#### 5.3.3.1 FDP_RIP.2 Full Residual Information Protection

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

### 5.3.4 Identification and authentication (FIA)

#### 5.3.4.1 FIA_AFL.1 Authentication Failure Handling

**FIA_AFL.1.1 Refinement:** The TSF shall detect when **an Administrator configurable positive integer of successive** unsuccessful authentication attempts occur related to **administrators attempting to authenticate remotely**.

**FIA_AFL.1.2** When the defined number of unsuccessful authentication attempts has been **met**, the TSF shall [prevent the offending remote administrator from successfully authenticating until *[an authorized administrator unlocks the locked user account]* is taken by a local Administrator].

#### 5.3.4.2 FIA_PMG_EXT.1 Password Management

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

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “ ” ‘ ` *(double or single quote/apostrophe), + (plus), - (minus), = (equal), , (comma), . (period), / (forward-slash), \ (back-slash), | (vertical-bar or pipe), : (colon), ; (semi-colon), < > (less-than, greater-than inequality signs), [ ] (square-brackets), { } (braces or curly-brackets ),? (question-mark), ^ (caret), _ (underscore), and ~ (tilde)]*;
2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;

#### 5.3.4.3 FIA_PSK_EXT.1 Extended: Pre-Shared Key Composition

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

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

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

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

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

#### 5.3.4.4 FIA_UIA_EXT.1 User Identification and Authentication

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

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

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

#### 5.3.4.5 FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

**FIA_UAU_EXT.2.1** The TSF shall provide a local password-based authentication mechanism, [*[support for RADIUS]*] to perform administrative user authentication.

#### 5.3.4.6 FIA_UAU.7 Protected Authentication Feedback

**FIA_UAU.7.1** The TSF shall provide only *obscured feedback* to the administrative user while the authentication is in progress at the local console.

#### 5.3.4.7 FIA_X509_EXT.1 Extended: X.509 Certificates

**FIA_X509_EXT.1.1** The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for IPsec and [no other protocols] connections.

**FIA_X509_EXT.1.2** The TSF shall store and protect certificate(s) from unauthorized deletion and modification.

**FIA_X509_EXT.1.3** The TSF shall provide the capability for authenticated Administrators to load X.509v3 certificates into the TOE for use by the security functions specified in this PP.

**FIA_X509_EXT.1.4** The TSF shall generate a Certificate Request Message as specified in RFC 2986 and be able to provide the following information in the request: public key, Common Name, Organization, Organizational Unit, and Country.

**FIA_X509_EXT.1.5** The TSF shall validate the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5759].

**FIA_X509_EXT.1.6** The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension is present and the cA flag is set to TRUE for all CA certificates.

**FIA_X509_EXT.1.7** The TSF shall not treat a certificate as a CA certificate if the basicConstraints extension is not present or the cA flag is not set to TRUE.

**FIA_X509_EXT.1.8** The TSF shall not establish an SA if a certificate or certificate path is deemed invalid.

**FIA_X509_EXT.1.9** The TSF shall not establish an SA if the distinguished name (DN) contained in a certificate does not match the expected DN for the entity attempting to establish a connection.

**FIA_X509_EXT.1.10** When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall, at the option of the administrator, establish an SA or disallow the establishment of an SA.

### 5.3.5 Security management (FMT)

#### 5.3.5.1 FMT_MOF.1 Management of Security Functions Behavior

**FMT_MOF.1.1 Refinement**: The TSF shall restrict the ability to enable, disable, determine and modify the behavior of all of the security functions of the TOE identified in this EP to an authenticated Administrator.

#### 5.3.5.2 FMT_MTD.1 Management of TSF Data (for general TSF data)

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

#### 5.3.5.3 FMT_SMF.1 Specification of Management Functions

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

- Ability to administer the TOE locally and remotely;
- Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;
- [
    - Configure Firewall rules,
    - Ability to configure the cryptographic functionality,
    - Ability to configure the IPsec functionality,
    - Ability to enable, disable, determine and modify the behavior of all the security functions of the TOE identified in this EP to the Administrator,
    - Ability to configure all security management functions identified in other sections of this EP]

#### 5.3.5.4 FMT_SMR.2 Restrictions on Security Roles

**FMT_SMR.2.1** The TSF shall maintain the roles:

- **Authorized Administrator.**

**FMT_SMR.2.2** The TSF shall be able to associate users with roles.

**FMT_SMR.2.3** The TSF shall ensure that the conditions

- **Authorized Administrator role shall be able to administer the TOE locally;**
- **Authorized Administrator role shall be able to administer the TOE remotely;**

are satisfied.

### 5.3.6 Protection of the TSF (FPT)

#### 5.3.6.1 FPT_ITT.1 Basic Internal TSF Data Transfer Protection

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

#### 5.3.6.2 FPT_SKP_EXT.1 Extended: Protection of TSF Data (for reading of all symmetric keys)

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

#### 5.3.6.3 FPT_APW_EXT.1 Extended: Protection of Administrator Passwords

**FPT_APW_EXT.1.1** The TSF shall store passwords in non-plaintext form.

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

#### 5.3.6.4 FPT_STM.1 Reliable time stamps

**FPT_STM.1.1** The TSF shall be able to provide reliable time stamps for its own use.

#### 5.3.6.5 FPT_TST_EXT.1: Extended: TSF Testing

**FPT_TST_EXT.1.1** The TSF shall run a suite of self tests during initial start-up (on power on) to demonstrate the correct operation of the TSF.

**FPT_TST_EXT.1.2** The TSF shall provide the capability to verify the integrity of stored TSF executable code when it is loaded for execution through the use of the TSF-provided cryptographic service specified in FCS_COP.1(2).

#### 5.3.6.6 FPT_TUD_EXT.1 Extended: Trusted Update

**FPT_TUD_EXT.1.1** The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.

**FPT_TUD_EXT.1.2** The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.

**FPT_TUD_EXT.1.3** The TSF shall provide a means to verify firmware/software updates to the TOE using a digital signature mechanism and [no other functions] prior to installing those updates.

#### 5.3.6.7 FPT_FLS.1 Fail Secure

**FPT_FLS.1.1 Refinement:** The TSF shall **shutdown** when the following types of failures occur: failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests.

### 5.3.7 TOE Access (FTA)

#### 5.3.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

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

- terminate the session]

after a Security Administrator-specified time period of inactivity.

#### 5.3.7.2 FTA_SSL.3(1) TSF-initiated Termination

**FTA_SSL.3.1(1) Refinement:** The TSF shall terminate **a remote** interactive session after a [*Security Administrator-configurable time interval of session inactivity*].

#### 5.3.7.1 FTA_SSL.3(2) TSF-initiated Termination

**FTA_SSL.3.1(2) Refinement:** The TSF shall terminate **a remote VPN client** session after a [*Security Administrator-configurable time interval of session inactivity*].

#### 5.3.7.2 FTA_SSL.4 User-initiated Termination

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

#### 5.3.7.3 FTA_TAB.1 Default TOE Access Banners

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

#### 5.3.7.4 FTA_TSE.1 TOE Session Establishment

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

#### 5.3.7.5 FTA_VCM_EXT.1 VPN Client Management

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

### 5.3.8 Trusted Path/Channels (FTP)

#### 5.3.8.1 FTP_ITC.1 Inter-TSF trusted channel

**FTP_ITC.1.1 Refinement:** The TSF shall **use IPsec, and** [no other protocols] to provide a **trusted** communication channel between itself and all authorized IT entities that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data **from disclosure and detection of modification of the channel data.**

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

**FTP_ ITC.1.3** The TSF shall initiate communication via the trusted channel for [

- *Audit server: transmit audit data via syslog over IPsec;*
- *Authentication server: authentication of TOE administrators using AAA servers including RADIUS over IPsec;*
- *Remote VPN peer using IPsec;*].

#### 5.3.8.2 FTP_TRP.1 Trusted Path

**FTP_TRP.1.1 Refinement:** The TSF shall **use** [SSH, TLS/HTTPS] provide **a trusted** communication path between itself and **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 detection of modification of the communicated data*.

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

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

### 5.3.9 Packeting Filtering (FPF)

#### 5.3.9.1 FPF_RUL_EXT.1 Packeting Filtering

**FPF_RUL_EXT.1.1** The TSF shall perform Packet Filtering on network packets processed by the TOE.

**FPF_RUL_EXT.1.2** The TSF shall process the following network traffic protocols:

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

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

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

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

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

and distinct interface.

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

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

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

**FPF_RUL_EXT.1.7** The TSF shall deny packet flow if a matching rule is not identified.

## 5.4 SFRs from the TFFWEP PP

### 5.4.1 Stateful Traffic Filtering (FFW)

#### 5.4.1.1 FFW_RUL_EXT.1 Stateful Traffic Filtering

**FFW_RUL_EXT.1.1** The TSF shall perform Stateful Traffic Filtering on network packets processed by the TOE.

**FFW_RUL_EXT.1.2** The TSF shall process the following network traffic protocols:

- Internet Control Message Protocol version 4 (ICMPv4)
- Internet Control Message Protocol version 6 (ICMPv6)
- Internet Protocol (IPv4)
- Internet Protocol version 6 (IPv6)
- Transmission Control Protocol (TCP)
- User Datagram Protocol (UDP)

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

- RFC 792 (ICMPv4)
- RFC 4443 (ICMPv6)
- RFC 791 (IPv4)
- RFC 2460 (IPv6)
- RFC 793 (TCP)
- RFC 768 (UDP).

**FFW_RUL_EXT.1.3** The TSF shall allow the definition of Stateful Traffic Filtering rules using the following network protocol fields:

- ICMPv4
  - Type
  - Code
- ICMPv6
  - Type
  - Code
- IPv4
  - Source address

  - Destination Address
  - Transport Layer Protocol
- IPv6
  - Source address
  - Destination Address
  - Transport Layer Protocol
- TCP
  - Source Port
  - Destination Port
- UDP
  - Source Port
  - Destination Port

**FFW_RUL_EXT.1.4** The TSF shall allow the following operations to be associated with Stateful Traffic Filtering rules: permit, deny, and log.

**FFW_RUL_EXT.1.5** The TSF shall allow the Stateful Traffic Filtering rules to be assigned to each distinct network interface.

**FFW_RUL_EXT.1.6** The TSF shall:

a) accept a network packet without further processing of Stateful Traffic Filtering rules if it matches an allowed established session for the following protocols: TCP, UDP, [ICMP] based on the following network packet attributes:

1. TCP: source and destination addresses, source and destination ports, sequence number, Flags;

2. UDP: source and destination addresses, source and destination ports;

3. [ICMP: source and destination addresses, [type, code]].

b) Remove existing traffic flows from the set of established traffic flows based on the following: [session inactivity timeout, completion of the expected information flow].

**FFW_RUL_EXT.1.7** The TSF shall be able to process the following network protocols:

1. FTP,

2. [no other protocols],

to dynamically define rules or establish sessions allowing network traffic of the following types:

- FTP: TCP data sessions in accordance with the FTP protocol as specified in RFC 959,
- [none]

**FFW_RUL_EXT.1.8** The TSF shall enforce the following default Stateful Traffic Filtering rules on all network traffic:

1. The TSF shall reject and be capable of logging packets which are invalid fragments;
2. The TSF shall reject and be capable of logging fragmented IP packets which cannot be re-assembled completely;

3. The TSF shall reject 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;
4. The TSF shall reject 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;
5. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a broadcast network;
6. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being on a multicast network;
7. The TSF shall reject and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;
8. The TSF shall reject and be capable of logging network packets where the source address of the network packet is a multicast;
9. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is a link-local address;
10. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as being an address "reserved for future use" as specified in RFC 5735 for IPv4;
11. The TSF shall reject and be capable of logging network packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" as specified in RFC 3513 for IPv6;
12. The TSF shall reject and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and
13. [*[Other traffic dropped by default and able to be logged:*

    i. *Slowpath Security Checks – The TSF shall reject and be capable of logging the detection of the following network packets:*
        1. *In routed mode when the ASA receives a through-the-box:*
            a. *L2 broadcast packet (MAC address FF:FF:FF:FF:FF:FF)*
            b. *IPv4 packet with destination IP address equal to 0.0.0.0*
            c. *IPv4 packet with source IP address equal to 0.0.0.0*
        2. *In routed or transparent mode when the ASA receives a through-the-box IPv4 packet with any of:*
            a. *first octet of the source IP address equal to zero*
            b. *network part of the source IP address equal to all 0's*
            c. *network part of the source IP address equal to all 1's*
            d. *source IP address host part equal to all 0's or all 1's*
            e. *source IP address and destination IP address are the same ("land.c" attack)*
        3. *IPv6 through-the-box packet with identical source and destination address.*

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

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

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

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

**FFW_RUL_EXT.1.9** When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall process the applicable Stateful Traffic Filtering rules (as determined in accordance with FFW_RUL_EXT.1.5) in the following order: administrator-defined.

**FFW_RUL_EXT.1.10** When FFW_RUL_EXT.1.6 or FFW_RUL_EXT.1.7 do not apply, the TSF shall deny packet flow if a matching rule is not identified.

## 5.5 TOE SFR Dependencies Rationale for SFRs Found in NDPP

The NDPPv1.1 contains all the requirements claimed in this Security Target. As such the dependencies are not applicable since the PP itself has been approved.

## 5.6 Security Assurance Requirements

### 5.6.1 SAR Requirements

The TOE assurance requirements for this ST are taken directly from the NDPP which are derived from Common Criteria Version 3.1, Revision 4. The assurance requirements are summarized in the table below.

**Table 14: Assurance Measures**

| Assurance Class | Components | Components Description |
|---|---|---|
| DEVELOPMENT | ADV_FSP.1 | Basic Functional Specification |
| GUIDANCE DOCUMENTS | AGD_OPE.1 | Operational User Guidance |
| | AGD_PRE.1 | Preparative User Guidance |
| LIFE CYCLE SUPPORT | ALC_CMC.1 | Labeling of the TOE |
| | ALC_CMS.1 | TOE CM Coverage |
| TESTS | ATE_IND.1 | Independent Testing - Conformance |
| VULNERABILITY ASSESSMENT | AVA_VAN.1 | Vulnerability Analysis |

### 5.6.2 Security Assurance Requirements Rationale

This Security Target claims conformance to the NDPP. This target was chosen to ensure that the TOE has a basic to moderate level of assurance in enforcing its security functions when instantiated in its intended environment which imposes no restrictions on assumed activity on applicable networks. The ST also claims conformance to the TFFWEP and VPNGWEP, which includes refinements to assurance measures for the SFRs defined in the TFFWEP and

VPNGWEP, including augmenting the vulnerability analysis (AVA_VAN.1) with specific vulnerability testing.

## 5.7 Assurance Measures

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

**Table 15: Assurance Measures**

| Component | How requirement will be met |
|---|---|
| ADV_FSP.1 | The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST. |
| AGD_OPE.1 | The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance. |
| AGD_PRE.1 | The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration. |
| ALC_CMC.1 | The Configuration Management (CM) document(s) describes how the consumer (end-user) of the TOE can identify the evaluated TOE (Target of Evaluation). The CM document(s), identifies the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked, how potential changes are incorporated, and the degree to which automation is used to reduce the scope for error. |
| ALC_CMS.1 | |
| ATE_IND.1 | Cisco provides the TOE for testing. |
| AVA_VAN.1 | Cisco provides the TOE for testing. |

# 6 TOE SUMMARY SPECIFICATION

## 6.1 TOE Security Functional Requirement Measures

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

**Table 16: How TOE SFRs Are Satisfied**

<table>
<tr><th>TOE SFRs</th><th>How the SFR is Satisfied</th></tr>
<tr><td colspan="2">Security Functional Requirements Drawn from NDPP</td></tr>
<tr><td>FAU_GEN.1</td><td>Shutdown and start-up of the audit functions are logged by events for reloading the TOE, and the events when the TOE comes back up. When audit is enabled, it is on whenever the TOE is on. Also, if logging is ever disabled, it is displayed in the ASDM Real-Time Log Viewer as a syslog disconnection and then a reconnection once it is re-established followed by an event that shows that the "logging enable" command was executed. See the table within this cell for other required events and rationale.<br>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:<br>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.<br>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:<br>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.<br>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<br>The following events are auditable by the TOE:
<table>
<tr><th>Auditable Event</th><th>Rationale</th></tr>
<tr><td>Modifications to the group of users that are part of the authorized administrator role.</td><td>All changes to the configuration (and hence all security relevant administrator actions) are logged when the logging level is set to at least the 'notifications' level. These changes would fall into the category of configuration changes. The identity of the administrator taking the action and the user being affected (assigned to the authorized administrator role) are both included within the event.</td></tr>
<tr><td>All use of the user identification mechanism.</td><td>Events will be generated for attempted identification/ authentication, and the username</td></tr>
</table>
</td></tr>
</table>

| TOE SFRs | How the SFR is Satisfied | |
|---|---|---|
| | | 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.<br>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 may need to be in each line of an access control list. The presumed addresses of the source and destination subjects are included in the event. |
| | 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. |
| | 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 | The connection to CA's or any other entity (e.g., CDP) for the purpose of certificate verification or revocation check is logged. |
| | Changes to the time. | Changes to the time are logged. |
| | 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. |

| TOE SFRs | How the SFR is Satisfied | |
|---|---|---|
| | Initiation of an update to the TOE. | TOE updates are logged as configuration changes. |
| | Termination of a remote session.<br>Note that the TOE does not support session locking, so there is no corresponding audit. | Termination of a remote session is logged as a terminated cryptographic path. This includes termination of remote VPN session as well. |
| | 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, and SSH. |
| | 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. |
| 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. | |
| FAU_STG_EXT.1 | The TOE can be configured export syslog records to an administrator-specified, external syslog server. The TOE can be configured to encrypt the communications with an external syslog server using IPsec.<br>If using TCP 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).<br>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. | |
| FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM_EXT.4, FCS_COP.1(1) through (4), and | In the TOE cryptographic functions are used to establish TLS, TLS, HTTPS, and SSH sessions, for IPsec traffic and authentication keys, and for IKE authentication and encryption keys.<br>Key generation for asymmetric keys on all models of the TOE implements ECDSA-based key establishment scheme as specified in NIST SP 800-56A | |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| FCS_RBG_EXT.1 | "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and RSA-based key establishment schemes as specified in NIST SP 800-56B "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" with key sizes greater than 112 bit key strength. The shall(not) and should(not) statements from NIST SP 800-56A and 800-56B that are implemented (or not) by the TOE are itemized in sections 8.3 and 8.4.<br><br>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 8.2. 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:<br><br>asa(config)#**crypto key zeroize rsa** [label key-pair-label] [default] [noconfirm]<br><br>asa(config)#**crypto key zeroize ec** [label key-pair-label]<br><br>The TOE supports AES-CBC, AES-GCM, and AES-GMAC, each with 128, 192, or 256-bit (as described in NIST SP 800-38A and 800-38D). The TOE uses a FIPS-validated implementation of AES with 128, 192, and 256 bit keys. Configuring the TOE software in or out of FIPS mode does not modify the TOE's use of the FIPS-validated AES.<br>• ASA 9.4(1) software: FIPS AES certification #3439<br>• CN1610 (ASA-5512-X, 5515-X, 5525-X)<br>• CN1620 (ASA-5545-X, 5555-X, 5585-X SSP10/20/40/60)<br><br>The TOE provides cryptographic signature services using RSA and ECDSA with key sizes (modulus) of 2048 bits, and 256 and 384 bits, respectively. For RSA, the key size is configurable down to 1024, but only 2048 key size is permitted in the evaluated configuration.<br>• ASA 9.4(1) software: FIPS RSA and RCDSA certifications #1760 & #693<br>• CN1610 (ASA-5512-X, 5515-X, 5525-X)<br>• CN1620 (ASA-5545-X, 5555-X, 5585-X SSP10/20/40/60)<br><br>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).<br>• ASA 9.4(1) software: FIPS SHA and HMAC certifications #2839 & #2188<br>• CN1610 (ASA-5512-X, 5515-X, 5525-X)<br>• CN1620 (ASA-5545-X, 5555-X, 5585-X SSP10/20/40/60)<br><br>Random number generation in the TOE uses different methods depending on the underlying hardware. The ASA multi-core platforms (5512-X, 5515-X, 5525-X, 5545-X, 5555-X, 5585-X, and the ASA-SM) use a NIST SP800-90 Hash DRBG |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | with SHA-512. The following Cavium NITROX (CN) security processors are used to generate entropy for random number generation:<br>• ASA 9.4(1) software: FIPS RBG certifications #838<br>• CN1610 (ASA-5512-X, 5515-X, 5525-X)<br>• CN1620 (ASA-5545-X, 5555-X, 5585-X SSP10/20/40/60) |
| FCS_HTTPS_EXT.1, and<br>FCS_TLS_EXT.1 | The TOE implements HTTP over TLS to support remote administration using ASDM. A remote administrator can connect over TLS to the TOE with their web browser and load the ASDM software from the ASDM. ASDM communicates with the TOE using HTTPS over TLS.<br>The TOE will support TLS v1.0 and TLSv1.2 connections with any of the following ciphersuites:<br>• TLS_RSA_WITH_AES_128_CBC_SHA<br>• TLS_RSA_WITH_AES_256_CBC_SHA<br>• TLS_DHE_RSA_WITH_AES_128_CBC_SHA<br>• TLS_DHE_RSA_WITH_AES_256_CBC_SHA<br>• TLS_RSA_WITH_AES_128_CBC_SHA256<br>• TLS_RSA_WITH_AES_256_CBC_ SHA256<br>• TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256<br>• TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256<br>• TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256<br>• TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 |
| FCS_IPSEC_EXT.1 | The IPsec implementation provides both VPN peer-to-peer and VPN client to TOE 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.<br>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.<br>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). The IKE protocols implement Peer Authentication using the RSA, ECDSA algorithm with |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | X.509v3 certificates, or pre-shared keys. IKE separates negotiation into two phases: phase 1 and phase 2. Phase 1 creates the first tunnel, which protects later IKE negotiation messages. The key negotiated in phase 1 enables IKE peers to communicate securely in phase 2. During Phase 2 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:<br>• The negotiation of mutually acceptable IPsec options between peers (including peer authentication parameters, either signature based or pre-shared key based),<br>• The establishment of additional Security Associations to protect packets flows using Encapsulating Security Payload (ESP), and<br>• 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<br><br>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), and using IKEv2, as specified for FCS_IKE_EXT.1, to establish security associations. NAT traversal is supported in IKEv2 by default.<br>The IKE Phase 1 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 SAs. Furthermore, the IKE SA lifetime limits can be configured so that no more than 200 MB of traffic can be exchanged for Phase 2 SAs. Administrators can require use of main mode by configuring the mode for each IPsec tunnel, as in the following examples:<br>**asa(config)#crypto map** *map-name seq-num* **set ikev2 phase1-mode main**<br>**asa(config)#crypto map** *map-name seq-num* **set security-association lifetime** {**seconds** *seconds* \| **kilobytes** *kilobytes*}<br>In the certified 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:<br>asa(config)#**crypto ipsec ikev2 ipsec-proposal** *proposal tag proposal_name*<br>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}<br>asa(config-ipsec-proposal)#**protocol esp integrity** {sha-1 \| sha-256 \| sha-384 \| sha-512 \| null}<br>**Note:** When AES-GCM, or AES-GMAC are used for encryption, the ESP integrity selection will be “null” because GCM and GMAC provide integrity.<br>The IKE protocols supported by the TOE implement the following DH groups: 14 (2048-bit MODP), 24 (2048-bit MODP with 256-bit POS), 19 (256-bit Random ECP), 20 (384-bit Random EC), and use the RSA and ECDSA algorithms for Peer |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | Authentication. The following command is used to specify the DH Group used for SAs:<br>asa(config)#**crypto ikev2 policy** *priority policy_index*<br>asa(config-ikev2-policy)#**group** {14 \| 19 \| 20 \| 24}<br>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 configured via the **crypto ipsec ikev2 sa-strength-enforcement** command.<br>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).<br>To configure an IKEv2 connection to use a RSA or ECDSA signature:<br>asa(config)#**tunnel-group** *name* **ipsec-attributes**<br>asa(config-tunnel-ipsec)#**ikev2** {local-authentication \| remote-authentication} **certificate** *trustpoint*<br>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 can be conditioned by a Password-Based Key Derivation Function 2 (PBKDF2) with SHA-256 hash. 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.<br>To configure an IKEv2 connection to use a pre-shared key:<br>asa(config)#**tunnel-group** *name* **ipsec-attributes**<br>asa(config-tunnel-ipsec)#**ikev2** {local-authentication \| remote-authentication} **pre-shared-key hex** *key-value*<br>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 sequence - the TOE attempts to match the packet to the access list (acl) specified in that entry. When a packet matches a permit entry in a particular access list, the method of security in the corresponding crypto map is applied. If the crypto map entry is tagged as ipsecisakmp, IPsec is triggered. The traffic matching the permit acls would then flow through the IPSec tunnel and be classified as PROTECTED. Traffic that does not match a permit acl in the crypto map, but that is not disallowed by other acls on the interface is allowed to BYPASS the tunnel. Traffic that does not match a permit acl and is also blocked by other non-crypto acls on the interface would be DISCARDED. |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| FCS_SSH_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 maximum number of failed authentication attempts limited to 3, and will be rekeyed upon request from the SSH client. SSH connections will be dropped if the TOE receives a packet larger than 65,535 bytes.<br><br>The TOE's implementation of SSHv2 supports:<br>• Public key algorithm RSA for signing and verification;<br>• Password-based authentication for administrative users;<br>• Encryption algorithms, AES-CBC-128, AES-CBC-256 to ensure confidentiality of the session;<br>• Hashing algorithm hmac-sha1[3] and hmac-sha1-96 to ensure the integrity of the session.<br>• Requiring use of DH group 14 by using the following command when enabling SSHv2 on an interface:<br><br>asa(config)#**ssh key-exchange dh-group14** {*ip_address mask* \| *ipv6_address*/*prefix*} *interface* |
| FDP_RIP.2 | The TOE ensures that packets transmitted from 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 both data plane traffic and administrative session traffic. |
| FIA_AFL.1 | The TOE provides the privileged administrator the ability to specify the maximum number of unsuccessful authentication attempts (between 1 and 25) before privileged administrator or non-privileged administrator is locked out.<br><br>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 through the administrative CLI. |
| 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 128 characters. Password composition rules specifying the types and number of required characters that comprise the password are settable by the Authorized Administrator. |

[3] When FIPS mode ('fips enable' command) is enabled, it will restrict what is allowed for SSH, including limiting HMAC to only hmac-sha1 and hmac-sha1-96.

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | Passwords can be configured with a maximum lifetime, configurable by the Authorized Administrator. New passwords can be required to contain a minimum of 4 character changes from the previous password. |
| FIA_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 that is input is conditioned prior to use via AES. |
| FIA_X509_EXT.1 | The TOE support X.509v3 certificates as defined by RFC 5280. 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.<br><br>The TOE can create a RSA or ECDSA public-private key pairs that can be used to generate a Certificate Signing Request (CSR). Through use of Simple Certificate Enrollment Protocol (SCEP), the TOE can: send the CSR to a Certificate Authority (CA) for the CA to sign and issue a certificate; and receive its X.509v3 certificate from the CA. Integrity of the CSR and certificate during transit are assured through use of digitally signatures (encrypting 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. 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 certificate is not accepted. |
| 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.<br><br>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 enable the user account. |
| FIA_UAU_EXT.2 | The TOE provides a local password based authentication mechanism as well as RADIUS authentication.<br><br>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 fail back to the |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | local user database if the remote authentication servers are inaccessible.<br><br>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).<br><br>The process for authentication is the same for administrative access whether administration is occurring via a directly connected console cable or remotely via SSHv2 or TLS. At initial login in the administrative user is prompted to provide a username. After the user provides the username, the user is prompted to provide the administrative password associated with the user account. The TOE then either grants administrative access (if the combination of username and password is correct) or indicates that the login was unsuccessful. The TOE does not provide indication of whether the username or password was the reason for an authentication failure. |
| FIA_UAU.7 | When a user enters their password at the local console, the TOE displays only '*' characters so that the user password is obscured. For remote session authentication, the TOE does not echo any characters as they are entered. |
| FMT_MOF.1 | The TOE restricts the ability to enable, disable, determine and modify the behavior of all of the security functions of the TOE to an authorized administrator. The TOE provides the ability for authorized administrators to access TOE data, such as audit data, configuration data, security attributes, information flow rules, and session thresholds. |
| FMT_MTD.1 | 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. Each of the predefined and administratively configured privilege level has delete 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" 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 ('ssh' and 'access list' commands), information flow rules ('access list' commands), modify the timestamp ('clock' commands), and specify limits for authentication failures ('aaa local authentication lockout') . These commands are not available outside of this |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | 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.<br><br>Note that the TOE does not provide services (other than connecting using SSH, HTTPS, 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 checksum (i.e., a published hash).<br><br>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. |
| FMT_SMR.2 | The TOE supports multiple levels of administrators, the highest of which is a privilege 15. In this evaluation privilege 15 would be the equivalent of the authorized administrator with full read-write access. Multiple level 15 administrators with individual usernames can be created.<br><br>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.<br><br>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.<br><br>'aaa authentication ssh console LOCAL' can be used to set the TOE to authenticate SSH users against the local database.<br><br>'aaa authorization exec' can be used to require re-authentication of users before they can get to EXEC mode.<br><br>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".<br><br>When command authorization has been enabled the default sets of privileges take effect at certain levels, and the levels become customizable.<br><br>• When "aaa authorization command LOCAL" has NOT been applied to the config:<br>o All usernames with level 2 and higher have the same full read-write access as if they had level 15 once their interactive session (CLI or ASDM) is effectively at level 2 or higher.<br>o Usernames with privilege levels 1 and higher can login to the CLI, and "enable" to their max privilege level (the level assigned to their username).<br>o Usernames with privilege levels 2-14 can login to ASDM, and have |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | full read-write access.<br>o Privilege levels cannot be customized.<br>• When "aaa authorization command LOCAL" has been applied to the config:<br>o Default command authorizations for privilege levels 3 and 5 take effect, where level 3 provides "Monitor Only" privileges, levels 4 and higher inherit privileges from level 3, level 5 provides "Read Only" privileges (a superset of Monitor Only privileges), and levels 6-14 inherit privileges from level 5.<br>o Privilege levels (including levels 3 and 5) can be customized from the default to add/remove specific privileges.<br>To display the set of privileges assigned to levels 3 or 5 (or any other privilege level), use "show running-config all privilege all", which shows all the default configuration settings that are not shown in the output of "show running-config all". |
| FPT_ITT.1 | Connection between TOE components occurs in three situations: when ASDM is used for remote administration; when clustering is configured; and when failover is configured. The ASDM-to-ASA connections (for remote administration) will use TLS/HTTPS. Failover connections can be made through use of a proprietary serial cable, or via network connection. The serial cable connection can be used in the evaluated configuration of the TOE as it is not a network-based connection, but network-based failover must remain disabled in the evaluated configuration because the proprietary AES-based encryption used for that link is not conformant to any of the encryption protocols allowed by the NDPP. |
| FPT_SKP_EXT.1 | The TOE stores all private keys in a secure directory (an 'opaque' virtual filesystem in RAM 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). |
| 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. |
| FPT_FLS.1 | 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.<br><br>Whenever a failure occurs within the TOE that results in the TOE ceasing operation, the TOE securely disables its interfaces to prevent the unintentional flow of any information to or from the TOE and reloads. So long as the failures persist, the TOE will continue to reload. This functionally prevents any failure from causing an unauthorized information flow. There are no failures that circumvent this protection. |
| FPT_STM.1 | The TOE provides a source of date and time information for the firewall, used in |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | audit timestamps, in validating service requests, and for tracking time-based actions related to session management including timeouts for inactive administrative sessions (FTA_SSL_EXT.*), and renegotiating SAs for IPsec tunnels (FCS_IPSEC_EXT.1). This function can only be accessed from within the configuration exec mode via the privileged mode of operation of the firewall. The clock function is reliant on the system clock provided by the underlying hardware.<br><br>This functionality can be set at the CLI using the 'clock' commands or in ASDM through the 'Configuration > Device Setup > System Time' page. The TOE can optionally be set to receive time from an NTP server.<br><br>All TOE models in the TOE contain a hardware-based real-time-clock (RTC) with battery-backup that maintains time in the event of power loss or reboot. 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. The TOE supports use of NTP version 3, which supports use of hashing to authenticate clock updates, but use of any hashing method in NTPv3 is outside the scope of this Common Criteria evaluation. |
| 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 install those updates.<br><br>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. Instructions for how to do this verification are provided in the administrator guidance for this evaluation. |
| FPT_TST_EXT.1 | The TOE run a suite of self-tests during initial start-up (power-on-self-tests or POST) to verify its correct operation. 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 software 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 software integrity testing verifies the image has not been tampered with or corrupted. If any of the self-tests fails, the TOE will cease operation. For more details, please see FPT_FLS.1. |
| FTA_SSL_EXT.1<br><br>FTA_SSL.3(1) | An administrator can configure maximum inactivity times for both local and remote administrative sessions. When a session is inactive (i.e., not session input) for the configured period of time the TOE will terminate the session, requiring the administrator to log in again to establish a new session when needed. |
| FTA_SSL.3(2) | 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-** |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | **timeout** or **default-idle-timeout** commands in the VPN configuration. |
| FTA_SSL.4 | An administrator is able to exit out of both local and remote administrative sessions, effectively terminating the session so it can't be re-used and will require authentication to establish a new session. |
| FTA_TAB.1 | The TOE provides administrators with the capability to configure advisory banner or warning message(s) that will be displayed prior to completion of the logon process at the local console or via any remote connection. |
| 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 of these properties to allow or deny access. In addition, the **vpn-access-hours** command can be used to restrict access based on date and time. |
| FTA_VCM_EXT.1 | The TOE provides the option to assign the remotely connecting VPN client an internal network IP address. The **ip-local-pool** command can be used to define the range of IP and IPv6 addresses to be available for use. |
| FTP_ITC.1 | The TOE uses IPsec to protect communications between itself and remote entities for the following purposes:<br>• The TOE protects transmission of audit records when sending syslog message to a remote audit server by transmitting the message over IPsec.<br>• Connections to authentication servers (AAA servers) can be protected via IPsec tunnels. Connections with AAA servers (via RADIUS) can be configured for authentication of TOE administrators.<br>• 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 TLS/HTTPS (for ASDM) to provide the trusted path (with protection from disclosure and modification) for all remote administration sessions. The TOE also supports tunneling the SSH and ASDM connections in IPsec VPN tunnels (peer-to-peer, or remote VPN client). |
| 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.<br>The TOE enforces information flow policies on network packets that are receive 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.<br>By implementing rules that defines the permitted flow of traffic between interfaces |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | of the TOE for unauthenticated traffic. These rules control whether a packet is transferred from one interface to another based on:<br>1. Presumed address of source<br>2. Presumed address of destination<br>3. Transport layer protocol (or next header in IPv6)<br>4. Service used (UDP or TCP ports, both source and destination)<br>5. Network interface on which the connection request occurs<br>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.<br>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.<br>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;<br>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;<br>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;<br>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;<br>These interfaces reject requests in which the subject specifies the route for information to flow when it is in route to its destination; and<br>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.<br>Otherwise, these interfaces pass traffic only when its source address matches the network interface originating the traffic through another network interface corresponding to the traffic's destination address.<br>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 |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | enforced on an interface. |
| Reproduced from the TFFWEP | |
| FFW_RUL_EXT.1.1<br>FFW_RUL_EXT.1.6 | 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, TCP and UDP port numbers, and ICMP source/destination addresses types and codes. 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.<br><br>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 8.1.<br><br>The proper session establishment "handshaking", and termination followed by the TOE is as defined in the following RFCs:<br>• RFC 792 (ICMPv4)<br>• RFC 4443 (ICMPv6)<br>• RFC 791 (IPv4)<br>• RFC 2460 (IPv6)<br>• TCP, RFC 793, section 2.7 Connection Establishment and Clearing<br>• UDP, RFC 768 (not applicable, UDP is a "stateless" protocol) |
| FFW_RUL_EXT.1.2,<br>FFW_RUL_EXT.1.3 | 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:<br>• 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);<br>• 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);<br>• Port numbers, and length as defined in RFC 768 (UDP); and<br>• Session identifiers, sequence numbers, types, and codes as defined in RFC 792 (ICMPv4), and RFC 4443 (ICMPv6).<br><br>Cisco confirms proper implementation of the RFCs through interoperability testing with Cisco and 3rd party products and through protocol compliant testing. |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | The TOE can also support deeper packet inspection and enforce additional RFC compliance beyond session management, but such traffic inspection functionality is not defined within the TFFWEP and is therefore beyond the scope of this CC certification. |
| FFW_RUL_EXT.1.4, FFW_RUL_EXT.1.5 | 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. 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.<br><br>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.:<br><br>asa(config)# **interface** gigabitethernet0/1<br><br>asa(config-if)# **nameif** inside<br><br>The interface types that can be assigned to an access-group are:<br>• Physical interfaces<br>o Ethernet<br>o GigabitEthernet<br>o TenGigabitEthernet<br>o Management<br>• Port-channel interfaces (designated by a port-channel number)<br>• Subinterface (designated by the subinterface number)<br><br>The default state of an interface depends on the type and the context mode:<br>• For the "system" context in single mode or multiple context mode, interfaces have the following default states:<br>o Physical interfaces = Disabled<br>o Subinterfaces = Enabled. However, for traffic to pass through the subinterface, the physical interface must also be enabled.<br>• 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. |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | 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.<br><br>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.<br><br>The Management 0/0 interface on the ASA 5512-X through ASA 5555-X has the following characteristics:<br><br>• No through traffic support<br>• No subinterface support<br>• No priority queue support<br>• No multicast MAC support<br>• If installed into a 5500-X, the SSP software module shares the Management 0/0 interface. Separate MAC addresses and IP addresses are used for the TOE and SSP module, and configuration of the SSP IP address must be performed within the SSP operating system. However, physical characteristics (such as enabling the interface) are configured on the TOE. |
| FFW_RUL_EXT.1.7 | The TOE supports numerous TCP and UDP protocols that require dynamic establishment of secondary network sessions including FTP. The TOE will manage establishment and teardown of the following protocols in accordance with the RFC for each protocol:<br><br>• FTP (File Transfer Protocol) is a TCP protocol supported in either active or passive mode:<br>o In active mode the client initiates the control session, and the server initiates the data session to a client port provided by the client;<br>o For active FTP to be allowed through the TOE, the firewall rules must explicitly permit the control session from the client to the server, and "inspect ftp" must be enabled. The TOE will then explicitly permit a control session to be initiated from the client to the server, and implicitly permit data sessions to be initiated from the server to the client while the control session is active.<br>o In passive (PASV) mode, the client initiates the control session, and the client also initiates the data session to a secondary port provided to the client by the server.<br>o 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 |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | destination port, and dynamically permit the data session. |
| FFW_RUL_EXT.1.8 | The TOE can be configured to implement default denial of various mal-formed packets/fragments, and other illegitimate network traffic, and can be configured to log that such packets/frames were dropped.<br><br>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.<br><br>asa(config)# **ip audit name** *name* {info \| attack} [action [alarm] [drop] [reset]]<br>asa(config)# **ip audit interface** *interface_name policy_name*<br><br>Example:<br><br>asa(config)# ip audit name ccpolicy1 attack action alarm reset<br>asa(config)# ip audit name ccpolicy2 info action alarm reset<br>asa(config)# ip audit interface outside ccpolicy1<br>asa(config)# ip audit interface inside ccpolicy2<br><br>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:<br><br>%ASA-4-4000*nn*: IPS:*number string* from *IP_address* to *IP_address* on interface *interface_name*<br><br>The following traffic will be denied by the TOE, and audit messages will be generated as indicated:<br><br>1. packets which are invalid fragments, including IP fragment attack<br><br>%ASA-2-106020: Deny IP teardrop fragment (size = *number*, offset = *number*) from *IP_address* to *IP_address*<br><br>%ASA-4-209004: Invalid IP fragment, size = *bytes* exceeds maximum size= *bytes*: src = *source_address*, dest = *dest_address*, proto = *protocol*, id = *number*<br><br>%ASA-4-402118: IPSEC: Received an *protocol* packet (SPI=*spi*, sequence number *seq_num*) from *remote_IP* (*username*) to *local_IP* containing an illegal IP fragment of length *frag_len* with offset *frag_offset*.<br><br>The following messages will be generated configured as described above.<br><br>%ASA-4-400007: IPS:1100 IP Fragment Attack from *IP_address* to *IP_address* on interface *interface_name*<br><br>%ASA-4-400009: IPS:1103 IP Overlapping Fragments (Teardrop) from *IP_address* to *IP_address* on interface *interface_name*<br><br>%ASA-4-400023: IPS:2150 Fragmented ICMP traffic from *IP_address* to *IP_address* on interface *interface_name*<br><br>%ASA-4-400025: IPS:2154 Ping of Death Attack from *IP_address* to *IP_address* on interface *interface_name* |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | *2.* fragmented IP packets which cannot be re-assembled completely;<br>%ASA-4-209003: Fragment database limit of *number* exceeded: src = *source_address*, dest = *dest_address*, proto = *protocol*, id = *number*<br>%ASA-4-209005: Discard IP fragment set with more than number elements: src = Too many elements are in a fragment set.<br>%ASA-4-423005: Dropped NBDGM *pkt_type_name* fragment with *error_reason_str* from *ifc_name*:*ip_address*/*port* to *ifc_name*:*ip_address*/*port*.<br>%ASA-4-507002: Data copy in proxy-mode exceeded the buffer limit<br>%ASA-7-715060: Dropped received IKE fragment. Reason: *reason*<br>`%ASA-7-715062: Error assembling fragments! Fragment numbers are non-continuous.`<br>*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;<br>%ASA-2-106016: Deny IP spoof from (*IP_address*) to *IP_address* on interface *interface_name.*<br>*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;<br>%ASA-2-106016: Deny IP spoof from (*IP_address*) to *IP_address* on interface *interface_name.*<br>This next message appears when Unicast RPF has been enabled with the **ip verify reverse-path** command.<br>%ASA-1-106021: Deny *protocol* reverse path check from *source_address* to *dest_address* on interface *interface_name*<br>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.<br>%ASA-1-106022: Deny *protocol* connection spoof from *source_address* to *dest_address* on interface *interface_name*<br>*5.* packets where the source address of the network packet is defined as being on a broadcast network;<br>%ASA-2-106016: Deny IP spoof from (*IP_address*) to *IP_address* on interface *interface_name.*<br>*6.* packets where the source address of the network packet is defined as being on a multicast network;<br>%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*<br>The preceding message will be generated when the rules listed below are configured without the “log” option. |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | %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<br>The preceding message will be generated when these rules are configured with the "log" option:<br>asa(config)#**object-group network** *grp_name*<br>asa(config-network-object-group)#**network-object** 224.0.0.0 255.0.0.0 #IPv4 multicast<br>asa(config-network-object-group)#**network-object** FF00::/8 #IPv6 multicast<br>asa(config)#**access-list** *acl-name* **extended deny ip** *grp-name* **any [log]**<br>asa(config)#**access-group in interface** *int-name*<br>*7.* packets where the source address of the network packet is defined as being a loopback address;<br>%ASA-2-106016: Deny IP spoof from (*IP_address*) to *IP_address* on interface *interface_name.*<br>The preceding message will be generated when no ACL has been defined to explicitly deny this traffic.<br>%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*<br>The preceding message will be generated when the rules listed below are configured without the "log" option.<br>%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<br>The preceding message will be generated when these rules are configured with the "log" option:<br>asa(config)#**object-group network** *grp_name*<br>asa(config-network-object-group)#**network-object** 127.0.0.0 255.0.0.0 #IPv4 loopback<br>asa(config-network-object-group)#**network-object** ::1/128 #IPv6 loopback<br>asa(config)#**access-list** *acl-name* **extended deny ip** *grp-name* **any [log]**<br>asa(config)#**access-group in interface** *int-name*<br>*8.* packets where the source address of the network packet is a multicast;<br>See item number 6.<br>*9.* packets where the source or destination address of the network packet is a link-local address;<br>%ASA-2-106016: Deny IP spoof from (*IP_address*) to *IP_address* on interface *interface_name.* |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | The preceding message will be generated when no ACL has been defined to explicitly deny this traffic.<br><br>%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*<br>The preceding message will be generated when the rules listed below are configured without the "log" option.<br><br>%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<br>The preceding message will be generated when these rules are configured with the "log" option:<br>asa(config)#**object-group network** *grp_name*<br>asa(config-network-object-group)#**network-object** 127.0.0.0 255.0.0.0 #IPv4 link-local<br>asa(config-network-object-group)#**network-object** FE80::/10 #IPv6 link-local<br>asa(config)#**access-list** *acl-name* **extended deny ip** *grp-name* **any [log]**<br>asa(config)#**access-list** *acl-name* **extended deny ip any** *grp-name* **[log]**<br>asa(config)#**access-group in interface** *int-name*<br><br>*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;<br><br>%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*<br>The preceding message will be generated when the rules listed below are configured without the "log" option.<br><br>%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<br>The preceding message will be generated when these rules are configured with the "log" option:<br>asa(config)#**object-group network** *grp_name*<br>asa(config-network-object-group)#**network-object** 192.0.0.0 255.0.0.0 #IPv4 reserved<br>asa(config-network-object-group)#**network-object** 240.0.0.0 128.0.0.0 #IPv4 reserved<br>asa(config)#**access-list** *acl-name* **extended deny ip** *grp-name* **any [log]**<br>asa(config)#**access-list** *acl-name* **extended deny ip any** *grp-name* **[log]**<br>asa(config)#**access-group in interface** *int-name*<br><br>*11.* packets where the source or destination address of the network packet is |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | defined as an "unspecified address" or an address "reserved for future definition and use" as specified in RFC 3513 for IPv6;<br><br>%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*<br>The preceding message will be generated when the rules listed below are configured without the "log" option.<br><br>%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<br>The preceding message will be generated when these rules are configured with the "log" option:<br>asa(config)#**object-group network** *grp_name*<br>asa(config-network-object-group)#**network-object** :: #IPv6 unspecified<br>asa(config-network-object-group)#**network-object** 0000::/8 #IPv6 reserved<br>asa(config)#**access-list** *acl-name* **extended deny ip** *grp-name* **any [log]**<br>asa(config)#**access-list** *acl-name* **extended deny ip any** *grp-name* **[log]**<br>asa(config)#**access-group in interface** *int-name*<br><br>*12.* Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified;<br><br>%ASA-6-106012: Deny IP from *IP_address* to *IP_address*, IP options *hex.*<br><br>The following messages will be generated when configured as described above.<br><br>%ASA-4-400001: IPS:1001 IP options-Record Packet Route from *IP_address* to *IP_address* on interface *interface_name*<br><br>%ASA-4-400004: IPS:1004 IP options-Loose Source Route from *IP_address* to *IP_address* on interface *interface_name*<br><br>%ASA-4-400006: IPS:1006 IP options-Strict Source Route from *IP_address* to *IP_address* on interface *interface_name*<br><br>*13.* By default, TOE will also drop (and is capable of logging) a variety of other IP packets with invalid content including:<br><br>• Invalid source and/or destination IP address including:<br>o source or destination is the network address (e.g. 0.0.0.0)<br>o source and destination address are the same (with or without the source and destination ports being the same)<br>o first octet of the source IP is equal to zero<br>o network part of the source IP is equal to all zeros or all ones<br>o host part of the source IP is equal to all zeros or all ones<br><br>• Invalid ICMP packets including: sequence number mismatch; invalid ICMP |

| TOE SFRs | How the SFR is Satisfied |
|---|---|
| | code, and ICMP responses unrelated to any established ICMP session |
| FFW_RUL_EXT.1.9 | TOE administrators have control over the sequencing of access control entries (ACEs) within an access control list (ACL) to be able to set the sequence in which ACEs are applied within any ACL. The entries within an ACL are always applied in a top-down sequence, and the first entry that matches the traffic is the one that's applied, regardless of whether there may be a more precise match for the traffic further down in the ACL. By changing the ordering/numbering of entries within an ACL, the administrator chances the sequence in which the entries are compared to network traffic flows. |
| FFW_RUL_EXT.1.10 | 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".<br><br>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. |

## 6.2 TOE Bypass and interference/logical tampering Protection Measures

The TOE consists of a hardware platform in which all operations in the TOE scope are protected from interference and tampering by untrusted subjects. All administration and configuration operations are performed within the physical boundary of the TOE. Also, all TSP enforcement functions must be invoked and succeed prior to functions within the TSC proceeding.

The TOE has been designed so that all locally maintained TSF data can only be manipulated via the secured management interface. There are no undocumented interfaces for managing the product.

All sub-components included in the TOE rely on the main chassis for power, memory management, and physical access control. In order to gain logical access any portion of the TOE, the Identification & Authentication mechanisms of the TOE must be invoked and succeed.

No processes outside of the TOE are allowed direct access to any TOE memory. The TOE only accepts traffic through legitimate TOE interfaces. Specifically, processes outside the TOE are not able to execute code on the TOE. None of these interfaces provide any access to internal TOE resources.

Finally, the TOE enforces information flow control policies through firewall rules and IPsec policies and applies network traffic security on its interfaces before traffic passes into or out of the TOE. The TOE controls every ingress and egress traffic flow. Traffic flows characterized as unauthorized are discarded and not permitted to circumvent the TSF. There are no unmediated traffic flows into or out of the TOE. During startup, the interfaces of the TOE are not operational (will not allow inbound, outbound, or through-the-box traffic) until after the Power-On Self-Test (POST) completes, and the startup configuration has been loaded. The loading of the startup configuration puts the TOE it its evaluated configuration all its administratively-defined traffic flow control policies (access-lists), and applies the access-lists to interfaces before the interfaces are enabled.

This design, combined with the fact that only an administrative user with the appropriate role may access the TOE security functions, provides a distinct protected domain for the TOE that is logically protected from interference and is not bypassable. When failures occur in hardware components, such as Network Interface Cards (NICs), the NIC may cease to forward traffic, but the software components of the TOE (such as traffic filtering, I&A, and auditing) continue to operated. When failures occur in software components, the TOE will crash (stopping all traffic flow), and attempt to reload to ensure that no security-relevant operations can be performed while the TOE security functions are not fully operational.

# 7 RATIONALE

This section describes the rationale for the Security Objectives and Security Functional Requirements as defined within the U.S. Government Protection Profiles for Security Requirements for Network Devices (NDPP), Traffic Filter Firewall Extended Package (TFFWEP), and VPN Gateway Extended Package (VPNGWEP).

## 7.1 Security objectives rationale

The security objectives rationale shows how the security objectives correspond to assumptions, threats, and organizational security policies and provide a justification of that tracing.

### 7.1.1 Tracing of security objectives to SPD

The tracing shows how the security objectives O.* and OE.* trace back to assumptions A.*, threats T.*, and organizational security policies OSP.* defined by the SPD.

**Table 17 Tracing of security objectives to SPD**

| | A.NO_GENERAL_PURPOSE | A.PHYSICAL | A.TRUSTED_ADMIN | A.CONNECTIONS | T.UNAUTHORIZED_ACCESS | T.UNAUTHORIZED_UPDATE | T.ADMIN_ERROR | T.UNDETECTED_ACTIONS | T.USER_DATA_REUSE | T.TSF_FAILURE | T.NETWORK_DISCLOSURE | T. NETWORK_ACCESS | T.NETWORK_MISUSE | T.NETWORK_DOS | T.REPLAY_ATTACK | T.DATA_INTEGRITY | T.UNAUTHORIZED_CONNECTI | T.HIJACKED_SESSION | T.UNPROTECTED_TRAFFIC | P.ACCESS BANNER |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **Reproduced from NDPP** | | | | | | | | | | | | | | | | | | | | |
| O.PROTECTED_ COMMUNICATIONS | | | | | | | | | | | | | | | | | | | | |
| O.VERIFIABLE_ UPDATES | | | | | | X | | | | | | | | | | | | | | |
| O.SYSTEM_ MONITORING | | | | | | | | X | | | | | X | | | | | | | |
| O.DISPLAY_ BANNER | | | | | | | | | | | | | | | | | | | | X |
| O.TOE_ ADMINISTRATION | | | | | | | X | | | | | | | | | | | | | |
| O.RESIDUAL_ INFORMATION_ CLEARING | | | | | | | | | X | | | | | | | | | | | |
| O.SESSION_LOCK | | | | | X | | | | | | | | | | | | | | | |
| O.TSF_SELF_TEST | | | | | | | | | | X | | | | | | | | | | |
| OE.NO_GENERAL_ PURPOSE | X | | | | | | | | | | | | | | | | | | | |
| OE.PHYSICAL | | X | | | | | | | | | | | | | | | | | | |
| OE.TRUSTED_ | | | X | | | | | | | | | | | | | | | | | |

| | A.NO_GENERAL_PURPOSE | A.PHYSICAL | A.TRUSTED_ADMIN | A.CONNECTIONS | T.UNAUTHORIZED_ACCESS | T.UNAUTHORIZED_UPDATE | T.ADMIN_ERROR | T.UNDETECTED_ACTIONS | T.USER_DATA_REUSE | T.TSF_FAILURE | T.NETWORK_DISCLOSURE | T. NETWORK_ACCESS | T.NETWORK_MISUSE | T.NETWORK_DOS | T.REPLAY_ATTACK | T.DATA_INTEGRITY | T.UNAUTHORIZED_CONNECTI | T.HIJACKED_SESSION | T.UNPROTECTED_TRAFFIC | P.ACCESS BANNER |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| ADMIN | | | | | | | | | | | | | | | | | | | | |
| **Reproduced from TFFWEP and VPNGWEP** | | | | | | | | | | | | | | | | | | | | |
| O.ADDRESS_ FILTERING | | | | | | | | | | | X | X | X | X | | | | | | |
| O.PORT_FILTERING | | | | | | | | | | | X | X | X | X | | | | | | |
| O.RELATED_ CONNECTION_ FILTERING | | | | | | | | | | | | X | | | | | | | | |
| O.STATEFUL_ INSPECTION | | | | | | | | | | | | | | X | | | | | | |
| O.AUTHENTICATIO N | | | | | | | | | | | X | | | | | | | | | |
| O.CRYPTOGRAPHI C_FUNCTIONS | | | | | | | | | | | | | | | X | X | | | | |
| O.FAIL_SECURE | | | | | | | | | | X | | | | | | | | | | |
| O.CLIENT_ESTABLI SHMENT_CONSTR AINTS | | | | | | | | | | | | | | | | | X | | | |
| O.REMOTE_SESSIO N_TERMINATION | | | | | | | | | | | | | | | | | | X | | |
| O.ASSIGNED_PRIV ATE_ADDRESS | | | | | | | | | | | | | | | | | | | X | |
| OE.CONNECTIONS | | | | X | | | | | | | | | | | | | | | | |

### 7.1.2 Justification of tracing

The justification demonstrates that the tracing of the security objectives to assumptions, threats, and OSPs is effective and all the given assumptions are upheld, all the given threats are countered, and all the given OSPs are enforced.

#### 7.1.2.1 Tracing of assumptions

**Table 18 Assumptions Rationale**

| Environment Objective | Rationale |
|---|---|
| OE.NO_GENERAL_PURPOSE | This security objective is necessary to address the assumption A.NO_GENERAL_PURPOSE by ensuring there are no general-purpose computing capabilities (e.g., the ability to execute arbitrary code or applications) capabilities on the TOE. |

| Environment Objective | Rationale |
|---|---|
| OE.PHYSICAL | This security objective is necessary to address the assumption A.PHYSICAL by ensuring the TOE and the data it contains is physically protected from unauthorized access. |
| OE.TRUSTED_ADMIN | This security objective is necessary to address the assumption A.TRUSTED_ADMIN by ensuring the administrators are non-hostile and follow all administrator guidance. |

#### 7.1.2.2 Tracing of threats and OSPs

**Table 19 Threat and OSP Rationale**

| Objective | Rationale |
|---|---|
| **Security Objectives Drawn from NDPP** | |
| O.PROTECTED_COMMUNICATIONS | This security objective is necessary to counter the threat: T.TRANSMIT to ensure the communications with the TOE is not compromised |
| O.VERIFIABLE_UPDATES | This security objective is necessary to counter the threat T.UNAUTHORIZED_UPDATE to ensure the end user has not installed a malicious update, thinking that it was legitimate. |
| O.SYSTEM_MONITORING | This security objective is necessary to counter the T.UNDETECTED_ACTIONS to ensure activity is monitored so the security of the TOE is not compromised. |
| O.DISPLAY_BANNER | This security objective is necessary to address the Organization Security Policy P.ACCESS_BANNER to ensure an advisory notice and consent warning message regarding unauthorized use of the TOE is displayed before the session is established. |
| O.TOE_ADMINISTRATION | This security objective is necessary to counter the T.ADMIN_ERROR that ensures actions performed on the TOE are logged so that indications of a failure or compromise of a TOE security mechanism are known and corrective actions can be taken. |
| O.RESIDUAL_INFORMATION_CLEARING | This security objective is necessary to counter the threat T.USER_DATA_REUSE so that data traversing the TOE could inadvertently be sent to a user other than that intended by the sender of the original network traffic. |
| O.SESSION_LOCK | This security objective is necessary to counter the threat: T.UNAUTHORIZED_ACCESS to ensure accounts cannot be compromised and used by an attacker that does not otherwise have access to the TOE. |
| O.TSF_SELF_TEST | This security objective is necessary to counter the threat T.TSF_FAILURE to ensure failure of mechanisms do not lead to a compromise in the TSF. |
| **Security Objectives Drawn from TFFWEP and VPNGWEP** | |
| O.ADDRESS_FILTERING | This security objective is necessary to counter the threats: T.NETWORK_DISCLOSURE, T. NETWORK_ACCESS, T.NETWORK_MISUSE, T.NETWORK_DOS to ensure the TOE will provide the means to filter and log network packets based on source and destination addresses. |

| Objective | Rationale |
|---|---|
| O.PORT_FILTERING | This security objective is necessary to counter the threats: T.NETWORK_DISCLOSURE, T. NETWORK_ACCESS, T.NETWORK_MISUSE, T.NETWORK_DOS to ensure the TOE will provide the means to filter and log network packets based on source and destination transport layer ports. |
| O.RELATED_CONNECTION_FILTERING | This security objective is necessary to counter the threat T.NETWORK_ACCESS to ensure for specific protocols, the TOE will dynamically permit a network packet flow in response to a connection permitted by the ruleset. |
| O.STATEFUL_INSPECTION | This security objective is necessary to address the threat T.NETWORK_DOS to ensure the TOE will determine if a network packet belongs to an allowed established connection before applying the ruleset. |
| O.AUTHENTICATION | This security objective is necessary to address the threat T.NETWORK_DISCLOSURE to ensure there is no unauthorized disclosure of information by authenticating the endpoint. |
| O.CRYPTOGRAPHIC_FUNCTIONS | This security objective is necessary to counter the threats T.REPLAY_ATTACK and T.DATA_INTEGRITY to ensure that strong cryptographic algorithms are used to protect data from corruption and reuse. |
| O.FAIL_SECURE | This security objective is necessary to address the threat T.TSF_FAILURE to ensure the TOE will shut down securely upon discovery of problems reported via the self-test mechanism. |
| O.CLIENT_ESTABLISHMENT_CONSTRAINTS | This security objective is necessary to address the threat T.UNAUTHORIZED_CONNECTIONS to ensure the VPN gateway only accepts remote client connections based on administrator defined attributes. |
| O.REMOTE_SESSION_TERMINATION | This security objective is necessary to address the threat T.HIJACKED_SESSION to ensure remote client sessions are terminated due to inactivity. |
| O.ASSIGNED_PRIVATE_ADDRESS | This security objective is necessary to address the threat T.UNPROTECTED_TRAFFIC to ensure assigning an IP address that the VPN Gateway can control. |

### 7.1.3 Security objectives conclusion

The tracing of the security objectives to assumptions, threats, and OSPs, and the justification of that tracing showed that all the given assumptions are upheld, all the given threats are countered, all the given OSPs are enforced, and the security problem as defined in the SPD is solved.

## 7.2 Rationale for requirements/TOE Objectives

The security requirements are derived according to the general model presented in Part 1 of the Common Criteria. Specifically, the tables below illustrate the mapping between the security requirements and the security objectives and the relationship between the threats, policies and IT security objectives. The functional and assurance requirements presented in the NDPP, TFFWEP, and VPNGWEP are mutually supportive and their combination meets the stated security objectives.

## 7.3 Rationale for TOE Security Objectives

**Table 20: SFR/Objectives Mappings**

| | O.PROTECTED_COMMUNICATIONS | O.VERIFIABLE_UPDATES | O.SYSTEM_MONITORING | O.DISPLAY_BANNER | O.TOE_ADMINISTRATION | O.RESIDUAL_INFORMATION_CLEARING | O.SESSION_LOCK | O.TSF_SELF_TEST | O.ADDRESS_FILTERING | O.PORT_FILTERING | O.RELATED_CONNECTION_FILTERING | O.STATEFUL_INSPECTION | O.AUTHENTICATION | O.CRYPTOGRAPHIC_FUNCTIONS | O.FAIL_SECURE | O.CLIENT_ESTABLISHMENT_CONSTRAINT | O.REMOTE_SESSION_TERMINATION | O.ASSIGNED_PRIVATE_ADDRESS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **SFRs drawn from the NDPP** | | | | | | | | | | | | | | | | | | |
| FAU_GEN.1 | | | X | | | | | | | | | | | | | | | |
| FAU_GEN.2 | | | X | | | | | | | | | | | | | | | |
| FAU_STG_EXT.1 | | | X | | | | | | | | | | | | | | | |
| FCS_CKM.1(1) | X | | | | | | | | | | | | | | | | | |
| FCS_CKM_EXT.4 | X | | | | | | | | | | | | | | | | | |
| FCS_COP.1(1) | X | | | | | | | | | | | | | X | | | | |
| FCS_COP.1(2) | X | X | | | | | | | | | | | | X | | | | |
| FCS_COP.1(3) | X | X | | | | | | | | | | | | X | | | | |
| FCS_COP.1(4) | X | | | | | | | | | | | | | X | | | | |
| FCS_HTTPS_EXT.1 | X | | | | | | | | | | | | | | | | | |
| FCS_RBG_EXT.1 | X | | | | | | | | | | | | | X | | | | |
| FCS_SSH_EXT.1 | X | | | | | | | | | | | | | | | | | |
| FCS_TLS_EXT.1 | X | | | | | | | | | | | | | | | | | |
| FDP_RIP.2 | | | | | | X | | | | | | | | | | | | |
| FIA_PMG_EXT.1 | | | | | | | | | | | | | X | | | | | |
| FIA_UIA_EXT.1 | | | | | X | | | | | | | | | | | | | |
| FIA_UAU_EXT.2 | | | | | X | | | | | | | | | | | | | |
| FIA_UAU.7 | | | | | X | | | | | | | | | | | | | |
| FMT_MTD.1 | | | | | X | | | | | | | | | | | | | |
| FMT_SMF.1 | | | | | X | | | | | | | | | | | | | |
| FMT_SMR.2 | | | | | X | | | | | | | | | | | | | |
| FPT_SKP_EXT.1 | X | | | | | | | | | | | | | | | | | |
| FPT_APW_EXT.1 | X | | | | | | | | | | | | | | | | | |
| FPT_STM.1 | | | X | | | | | | | | | | | | | | | |
| FPT_TUD_EXT.1 | | X | | | | | | | | | | | | | | | | |
| FPT_TST_EXT.1 | | | | | | | | X | | | | | | | | | | |
| FTA_SSL_EXT.1 | | | | | | | X | | | | | | | | | | | |
| FTA_SSL.3(1) | | | | | | | X | | | | | | | | | | | |
| FTA_SSL.4 | | | | | | | X | | | | | | | | | | | |
| FTA_TAB.1 | | | | X | | | | | | | | | | | | | | |
| FTP_ITC.1 | X | | | | | | | | | | | | X | | | | | |

| | O.PROTECTED_COMMUNICATIONS | O.VERIFIABLE_UPDATES | O.SYSTEM_MONITORING | O.DISPLAY_BANNER | O.TOE_ADMINISTRATION | O.RESIDUAL_INFORMATION_CLEARING | O.SESSION_LOCK | O.TSF_SELF_TEST | O.ADDRESS_FILTERING | O.PORT_FILTERING | O.RELATED_CONNECTION_FILTERING | O.STATEFUL_INSPECTION | O.AUTHENTICATION | O.CRYPTOGRAPHIC_FUNCTIONS | O.FAIL_SECURE | O.CLIENT_ESTABLISHMENT_CONSTRAINT | O.REMOTE_SESSION_TERMINATION | O.ASSIGNED_PRIVATE_ADDRESS |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| FTP_TRP.1 | X | | | | | | | | | | | | | | | | | |
| **SFRs drawn from the TFFWEP** | | | | | | | | | | | | | | | | | | |
| FFW_RUL_EXT.1 | | | | | | | | | X | X | X | X | | | | | | |
| **SFRs drawn from the VPNGWEP** | | | | | | | | | | | | | | | | | | |
| FCS_CKM.1(2) | X | | | | | | | | | | | | | | | | | |
| FCS_IPSEC_EXT.1 | X | | | | | | | | | | | | X | X | | | | |
| FIA_AFL.1 | | | | | X | | | | | | | | | | | | | |
| FIA_PSK_EXT.1 | X | | | | | | | | | | | | | | | | | |
| FIA_X509_EXT.1 | X | | | | | | | | | | | | | | | | | |
| FMT_MOF.1 | | | | | X | | | | | | | | | | | | | |
| FPF_RUL_EXT.1 | | | X | | | | | | X | X | | | | | | | | |
| FPT_FLS.1 | | | | | | | | | | | | | | | X | | | |
| FTA_SSL.3(2) | | | | | | | | | | | | | | | | | X | |
| FTA_TSE.1 | | | | | | | | | | | | | | | | X | | |
| FTA_VCM_EXT.1 | | | | | | | | | | | | | | | | | | X |

The inspection of Table 20 shows that:

- Each SFR traces back to at least one security objective;
- Each security objective for the TOE has at least one SFR tracing to it.

#### 7.3.1.1 Justification of SFR tracing

The justification demonstrates that the SFRs address all security objectives of the TOE.

**Table 21 SFR Tracing Justification**

| Objective | Rationale |
|---|---|
| **Objectives Drawn from NDPP** | |
| O.PROTECTED_COMMUNICATIONS | The SFRs, FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_HTTPS_EXT.1, FCS_IPSEC_EXT.1, FCS_SSH_EXT.1, FCS_RBG_EXT.1, FCS_TLS_EXT.1, FCS_HTTPS_EXT.1, FPT_PTD_EXT.1(2), FPT_RPL.1, FTP_ITC.1, and FTP_TRP.1 meet this objective by ensuring the communications between the TOE and endpoints are secure by implementing the security algorithms and protocols as |

| Objective | Rationale |
|---|---|
| | defined in the SFRs and as specified by the RFCs. |
| O.VERIFIABLE_UPDATES | The SFRs, FPT_TUD_EXT.1, FCS_COP.1(2), and FCS_COP.1(3) meet this objective by ensuring the update was downloaded via secure communications, is from a trusted source, and the update can be verified by cryptographic mechanisms prior to installation. |
| O.SYSTEM_MONITORING | The SFRs, FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1, FPT_STM.1, FPF_RUL_EXT.1 meet this objective by auditing actions on the TOE. The audit records identify the user associated with the action/event, whether the action/event was successful or failed, the type of action/event, and the date/time the action/event occurred. The audit logs are transmitted securely to a remote syslog server. If connectivity to the remote syslog server is lost, the TOE will block new permit actions. |
| O.DISPLAY_BANNER | The SFR, FTA_TAB.1 meets this objective by displaying an advisory notice and consent warning message regarding unauthorized use of the TOE. |
| O.TOE_ADMINISTRATION | The SFRs, FIA_UIA_EXT.1, FIA_PMG_EXT.1, FIA_UAU_EXT.5, FIA_UAU.6, FIA_UAU.7, FMT_MOF.1, FMT_MSA.2, FMT_MSA.3, FMT_MTD.1, FMT_SMF.1, FMT_SFR.1, FPT_PTD_EXT.1(1), FTA_SSL_EXT.1, FTA_SSL.3, and FIA_AFL.1 meet this objective by ensuring the TOE supports a password-based authentication mechanism with password complexity enforcement such as, strong passwords, password life-time constraints, providing current password when changing the password, obscured password feedback when logging in, and passwords are not stored in plaintext. The objective is further met by ensuring restrictive default values are enforced on the SFPs (authorization and flow control), that only authorized administrators to override the default values, that the TOE provides the management and configuration features to securely manage the TOE and that those functions are restricted to the authorized administrator, and the implementation of session termination after an administrative configurable inactivity time period whereas the user must be re-authenticated. In addition, the TOE provides the ability for an authorized administrator to exit, unlock, or logoff an administrator session. The TOE must also protects itself against attempts by unauthorized users to bypass, deactivate, or tamper with TOE security functions.<br><br>The TOE will provide the authorized administrators the capability to review Audit data. Security relevant events must be available for review by authorized administrators as provided by FAU_SAR.1. The TOE does not have an interface to modify audit records, though there is an interface available for the authorized administrator to delete audit data stored locally on the TOE as provided by FAU_STG.1. |
| O.RESIDUAL_INFORMATION_CLEARING | The SFR, FDP_RIP.2 meets this objective by ensuring no left over user data from the previous transmission is included in the network traffic. |
| O.RESOURCE_AVAILABILITY | The SFR, FRU_RSA.1 meets this objective by limiting the number of amount of exhaustible resources, such the number of concurrent administrative sessions. |
| O.SESSION_LOCK | The SFRs, FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4 meet this objective by terminating a session due to meeting/exceeding the inactivity time limit. In addition, the TOE allows an authorized administrator to exit or logoff an administrator session. |
| O.TSF_SELF_TEST | The SFR, FPT_TST_EXT.1 meets this objective by performing self-test to ensure the TOE is operating correctly and all functions are available and enforced. |

| Objective | Rationale |
|---|---|
| **Reproduced from the TFFWEP and VPNGWEP** | |
| O.ADDRESS_FILTERING | The SFRs, FFW_RUL_EXT.1 and FPF_RUL_EXT.1 meet this objective by providing the means to filter and log network packets based on source and destination IP addresses. |
| O.PORT_FILTERING | The SFRs, FFW_RUL_EXT.1 and FPF_RUL_EXT.1 meet this objective by providing the means to filter and log network packets based on source and destination transport layer ports. |
| **Reproduced from the TFFWEP** | |
| O.STATEFUL_INSPECTION | The SFR, FFW_RUL_EXT.1 meets this objective by determine if a network packet belongs to an allowed established connection before applying the ruleset. |
| O.RELATED_CONNECTION_FILTERING | The SFR, FFW_RUL_EXT.1 meets this objective by dynamically permitting a network packet flow in response to a connection permitted by the ruleset. |
| **Reproduced from the VPNGWEP** | |
| O.AUTHENTICATION | The SFRs, FTP_ITC.1 and FCS_IPSEC_EXT.1, meet this objective by using IKE/IPsec to authenticate each other to ensure they are communicating with an authorized external IT entity. |
| O.CRYPTOGRAPHIC_FUNCTIONS | The SFRs, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1, and FCS_IPSEC_EXT.1, meet this objective by implementing cryptographic algorithms and protocols to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE. |
| O.FAIL_SECURE | The SFR, FPT_FLS.1, meets this objective by shutting down the TOE securely when malfunctions are detected or the integrity of the TOE's software is compromised. |
| O.CLIENT_ESTABLISHMENT_CONSTRAINTS | The SFR, FTA_TSE.1, meets this objective by restricting the VPN client connections based on attributes defined by the authorized administrators. |
| O.REMOTE_SESSION_TERMINATION | The SFR, FTA_SSL.3(2), meets this objective by terminating VPN client connections if a period of inactivity configured by authorized administrators has been met. |
| O.ASSIGNED_PRIVATE_ADDRESS | The SFR, FTA_VCM_EXT.1, meets this objective by assigning IP addresses that the TOE controls. |

# 8 SUPPLEMENTAL TOE SUMMARY SPECIFICATION INFORMATION

## 8.1 Tracking of Stateful Firewall Connections

### 8.1.1 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.

### 8.1.2 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-ASA connections. To include connections to/from the ASA itself in the command output, add the **all** keyword, "**show conn all**".

**Table 22: Syntax Description**

| | |
|---|---|
| **address** | (Optional) Displays connections with the specified source or destination IP address. |
| **all** | (Optional) Displays connections that are to the device or from the device, in addition to through-traffic connections. |
| **count** | (Optional) Displays the number of active connections. |
| *dest_ip* | (Optional) Specifies the destination IP address (IPv4 or IPv6). To specify a range, separate the IP addresses with a dash (-). For example:<br>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:<br>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:<br>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:<br>1000-2000 |
| **state** *state_type* | (Optional) Specifies the connection state type. See Table 46-5 for a list of the keywords available for connection state types. |
| **user** [*domain_nickname*\] *user_name* | (Optional) Specifies that all connections displayed belong to the specified user. When you do not include the *domain_nickname* argument, the TOE displays information for the user in the default domain. |
| **user-group** [*domain_nickname*\\] *user_group_name* | (Optional) Specifies that all connections displayed belong to the specified user group. When you do not include the *domain_nickname* argument, the TOE displays information for the user group in the default domain. |
| **user-identity** | (Optional) Specifies that the TOE display all connections for the Identity Firewall feature. When displaying the connections, the TOE displays the user name and IP address when it identifies a matching user. Similarly, the TOE displays the host name and an IP address when it identifies a matching host. |

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

**Table 23: Connection State Types**

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

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

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

**Table 24: Connection State Flags**

| Flag | Description |
|---|---|
| a | awaiting outside ACK to SYN |
| A | awaiting inside ACK to SYN |
| b | TCP state bypass. By default, all traffic that passes through the Cisco Adaptive Security Appliance (ASA) 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 ASA 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 ASA preallocates an outside back connection with this flag set. If the inside client attempts to connect back to the server, then the ASA 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<br>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<br>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 ASA preallocates an inside back connection with this flag set. If the outside server attempts to connect back to the client, then the ASA 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<br>If UDP: UDP RPC2<br>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<br>For a UDP connection, the value t indicates that it will timeout after one minute. |
| T | SIP connection<br>For UDP connections, the value T indicates that the connection will timeout according to the value specified using the “timeout sip” command. |
| U | up |
| V | VPN orphan |
| W | WAAS |
| X | Inspected by the service module, such as a CSC SSM. |
| y | For clustering, identifies a backup owner flow. |
| Y | For clustering, identifies a director flow. |
| z | For clustering, identifies a forwarder flow. |
| Z | Cloud Web Security |

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

DNS connection being reset by a new DNS session. This is due to the nature of the shared DNS connection and is by design.

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

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

**Table 25: TCP connection directionality flags**

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

### 8.1.3 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

## 8.2 Key Zeroization

The following table describes the key zeroization referenced by FCS_CKM_EXT.4 provided by the TOE.

**Table 26: TOE Key Zeroization**

| Critical Security Parameters (CSPs) | Zeroization Cause and Effect |
|---|---|
| Diffie-Hellman Shared Secret | Automatically zeroized after completion of DH exchange, by calling a specific API within the two crypto modules, when module is shutdown, or reinitialized.<br>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.<br>Overwritten with: 0x00 |
| skeyid | Session Encryption Key and IKE Session Authentication Key. Automatically zeroized after IKE session terminated.<br>Overwritten with: 0x00 |
| skeyid_d | Session Encryption Key and IKE Session Authentication Key. Automatically zeroized after IKE session terminated.<br>Overwritten with: 0x00 |
| IKE Session Encryption Key | Session Encryption Key and IKE Session Authentication Key. Automatically zeroized after IKE session terminated.<br>Overwritten with: 0x00 |
| IKE Session Authentication Key | Session Encryption Key and IKE Session Authentication Key. Automatically zeroized after IKE session terminated.<br>Overwritten with: 0x00 |
| ISAKMP Preshared | Zeroized using the following command:<br>**# no crypto isakmp key**<br>Overwritten with: 0x00 |
| IKE RSA and ECDSA Private Keys | Automatically overwritten when a new key is generated or zeroized using the following commands:<br>**# crypto key zeroize rsa**<br>**# crypto key zeroize ec**<br>Overwritten with: 0x00 |

| Critical Security Parameters (CSPs) | Zeroization Cause and Effect |
|---|---|
| IPsec Encryption Key | Automatically zeroized when IPsec session terminated.<br>Overwritten with: 0x00 |
| IPsec Authentication Key | Automatically zeroized when IPsec session terminated.<br>Overwritten with: 0x00 |
| RADIUS Secret | Zeroized using the following command:<br>**# no radius-server key**<br>Overwritten with: 0x00 |
| SSH Private Key | Automatically zeroized upon generation of a new key<br>Overwritten with: 0x00 |
| SSH Session Key | Automatically zeroized when the SSH session is terminated.<br>Overwritten with: 0x00 |
| All CSPs | Zeroized on-demand on all file systems via the “erase” command. |

## 8.3 NIST Special Publication 800-56A

The TOE is compliant with NIST SP 800-56A as described in Table 28 below.

**Table 27 800-56A Compliance**

| Section | Exceptions to Shall/Shall Not Statement(s) | Should (Not) Statements[4] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| 5.1 Cryptographic Hash Functions | None. | None. | Yes | N/A |
| 5.2 Message Authentication Code (MAC) Algorithm | None. | None. | Yes | N/A |

[4] This column does not included “should/should not” statements that relate to the “owner”, “recipient”, “application”, or “party” as they are outside of the scope of the TOE.

| Section | Exceptions to Shall/Shall Not Statement(s) | Should (Not) Statements[4] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| 5.2.1 MacTag Computation | None. | None. | Yes | N/A |
| 5.2.2 MacTag Checking | None. | None. | Yes | N/A |
| 5.2.3 Implementation Validation Message | None. | None. | Yes | N/A |
| 5.3 Random Number Generation | None. | None. | Yes | N/A |
| 5.4 Nonces | None. | "a random nonce **should** be used" is met by the TOE. | Yes | N/A |
| 5.5 Domain Parameters | None. | None. | Yes | N/A |
| 5.5.1 Domain Parameter Generation | None. | "If the appropriate security strength does not have an FFC parameter set, then Elliptic Curve Cryptography should be used" FFC parameter is set, so this is N/A | Yes | N/A |
| 5.5.1.1 FFC Domain Parameter Generation | None. | None. | Yes | N/A |
| 5.5.1.2 ECC Domain Parameter Generation | None. | None. | Yes | N/A |
| 5.5.2 Assurances of Domain Parameter Validity | None. | None. | Yes | N/A |
| 5.5.3 Domain Parameter Management | None. | None. | Yes | N/A |
| 5.6 Private and Public Keys | None. | None. | Yes | N/A |
| 5.6.1 Private/Public Key Pair Generation | None. | None. | Yes | N/A |
| 5.6.1.1 FFC Key Pair | For the FFC schemes, each static and | None. | No | Prime number |

| Section | Exceptions to Shall/Shall Not Statement(s) | Should (Not) Statements[4] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| Generation | ephemeral private key and public key shall be generated using an Approved method and the selected valid domain parameters (p, q, g{, SEED, pgenCounter}) (see Appendix B of FIPS 186-3). | | | generation is done as described in SP 800-90 |
| | Each private key shall be unpredictable and shall be generated in the range [1, q-1] using an Approved random bit generator. | None. | Yes | N/A |
| 5.6.1.2 ECC Key Pair Generation | For the ECC schemes, each static and ephemeral private key d and public key Q shall be generated using an Approved method and the selected domain parameters (q, FR, a, b{, SEED}, G, n, h). | None. | No | This section specifies that C must be less than N-2. However, the implementation makes the calculation on C must be less then N. |
| | Each private key, d, shall be unpredictable and shall be generated in the range [1, n-1] using an Approved random bit generator. | None. | Yes | N/A |
| 5.6.2 Assurances of the Arithmetic Validity of a Public Key | None. | None. | Yes | N/A |
| 5.6.2.1 Owner Assurances of Static Public Key Validity | None. | None. | Yes | N/A |
| 5.6.2.2 Recipient Assurances of Static Public Key Validity | None. | None. | Yes | N/A |
| 5.6.2.3 Recipient Assurances of Ephemeral Public Key Validity | None. | None. | Yes | N/A |
| 5.6.2.4 FFC Full Public Key Validation Routine | None. | None. | Yes | N/A |
| 5.6.2.5 ECC Full | None. | None. | Yes | N/A |

| Section | Exceptions to Shall/Shall Not Statement(s) | Should (Not) Statements[4] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| Public Key Validation Routine | | | | |
| 5.6.2.6 ECC Partial Public Key Validation Routine | None. | None. | Yes | N/A |
| 5.6.3 Assurances of the Possession of a Static Private Key | None. | None. | Yes | N/A |
| 5.6.3.1 Owner Assurances of Possession of a Static Private Key | None. | None. | Yes | N/A |
| 5.6.3.2 Recipient Assurance of Owner's Possession of a Static Private Key | None. | None. | Yes | N/A |
| 5.6.3.2.1 Recipient Obtains Assurance Directly from the Claimed Owner | None. | None. | Yes | N/A |
| 5.6.4 Key Pair Management | None. | None. | Yes | N/A |
| 5.6.4.1 Common Requirements on Static and Ephemeral Key Pairs | None. | None. | Yes | N/A |
| 5.6.4.2 Specific Requirements on Static Key Pairs | None. | None. | Yes | N/A |
| 5.6.4.3 Specific Requirements on Ephemeral Key Pairs | None. | "An ephemeral key pair should be generated as close to its time of use as possible" | Yes | N/A |
| 5.7 DLC Primitives | None. | None. | Yes | N/A |
| 5.7.1 Diffie-Hellman Primitives | None. | None. | Yes | N/A |

| Section | Exceptions to Shall/Shall Not Statement(s) | Should (Not) Statements[4] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| 5.7.1.1 Finite Field Cryptography Diffie-Hellman (FFC DH) Primitive | None. | None. | Yes | N/A |
| 5.7.1.2 Elliptic Curve Cryptography Cofactor Diffie-Hellman (ECC-CDH) Primitive | None. | None. | Yes | N/A |
| 5.7.2 MQV Primitives | None. | None. | Yes | N/A |
| 5.7.2.1 Finite Field Cryptography MQV (FFC MQV) Primitive | None. | None. | Yes | N/A |
| 5.7.2.1.1 MQV2 Form of the FFC MQV Primitive | None. | None. | Yes | N/A |
| 5.7.2.1.2 MQV1 Form of the FFC MQV Primitive | None. | None. | Yes | N/A |
| 5.7.2.2 ECC MQV Associate Value Function | None. | None. | Yes | N/A |
| 5.7.2.3 Elliptic Curve Cryptography MQV (ECC MQV) Primitive | None. | None. | Yes | N/A |
| 5.7.2.3.1 Full MQV Form of the ECC MQV Primitive | None. | None. | Yes | N/A |
| 5.7.2.3.2 One-Pass Form of the ECC MQV Primitive | None. | None. | Yes | N/A |
| 5.8 Key Derivation Functions for Key Agreement Schemes | None. | None. | Yes | N/A |
| 5.8.1 Concatenation Key Derivation | None. | None. | Yes | N/A |

| Section | Exceptions to Shall/Shall Not Statement(s) | Should (Not) Statements[4] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| Function (Approved Alternative 1) | | | | |
| 5.8.2 ASN.1 Key Derivation Function (Approved Alternative 2) | None. | None. | Yes | N/A |
| 6. Key Agreement | None. | None. | Yes | N/A |
| 6.1 Schemes Using Two Ephemeral Key Pairs, C(2) | None. | None. | Yes | N/A |
| 6.1.1 Each Party Has a Static Key Pair and Generates an Ephemeral Key Pair, C(2, 2) | None. | None. | Yes | N/A |
| 6.1.1.1 dhHybrid1, C(2, 2, FFC DH) | None. | None. | Yes | N/A |
| 6.1.1.2 Full Unified Model, C(2, 2, ECC CDH) | None. | None. | Yes | N/A |
| 6.1.1.3 MQV2, C(2, 2, FFC MQV) | None. | None. | Yes | N/A |
| 6.1.1.4 Full MQV, C(2, 2, ECC MQV) | None. | None. | Yes | N/A |
| 6.1.1.5 Rationale for Choosing a C(2, 2) Scheme | None. | None. | Yes | N/A |
| 6.1.2 Each Party Generates an Ephemeral Key Pair; No Static Keys are Used, C(2, 0) | None. | None. | Yes | N/A |
| 6.1.2.1 dhEphem, C(2, 0, FFC DH) | None. | None. | Yes | N/A |
| 6.1.2.2 Ephemeral Unified Model, C(2, 0, ECC CDH) | None. | None. | Yes | N/A |

| Section | Exceptions to Shall/Shall Not Statement(s) | Should (Not) Statements[4] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| 6.1.2.3 Rationale for Choosing a C(2, 0) Scheme | None. | None. | Yes | N/A |
| 6.2 Schemes Using One Ephemeral Key Pair, C(1) | None. | None. | Yes | N/A |
| 6.2.1 Initiator Has a Static Key Pair and Generates an Ephemeral Key Pair; Responder Has a Static Key Pair, C(1, 2) | None. | None. | Yes | N/A |
| 6.2.1.1 dhHybridOneFlow, C(1, 2, FFC DH) | None. | None. | Yes | N/A |
| 6.2.1.2 One-Pass Unified Model, C(1, 2, ECC CDH) | None. | None. | Yes | N/A |
| 6.2.1.3 MQV1, C(1, 2, FFC MQV) | None. | None. | Yes | N/A |
| 6.2.1.4 One-Pass MQV, C(1, 2, ECC MQV) | None. | None. | Yes | N/A |
| 6.2.1.5 Rationale for Choosing a C(1, 2) Scheme | None. | None. | Yes | N/A |
| 6.2.2 Initiator Generates Only an Ephemeral Key Pair; Responder Has Only a Static Key Pair, C(1, 1) | None. | None. | Yes | N/A |
| 6.2.2.1 dhOneFlow, C(1, 1, FFC DH) | None. | None. | Yes | N/A |
| 6.2.2.2 One-Pass Diffie-Hellman, C(1, 1, ECC CDH) | None. | None. | Yes | N/A |
| 6.2.2.3 Rationale in | None. | None. | Yes | N/A |

| Section | Exceptions to Shall/Shall Not Statement(s) | Should (Not) Statements[4] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| Choosing a C(1, 1) Scheme | | | | |
| 6.3 Scheme Using No Ephemeral Key Pairs, C(0, 2) | None. | None. | Yes | N/A |
| 6.3.1 dhStatic, C(0, 2, FFC DH) | None. | None. | Yes | N/A |
| 6.3.2 Static Unified Model, C(0, 2, ECC CDH) | None. | None. | Yes | N/A |
| 6.3.3 Rationale in Choosing a C(0, 2) Scheme | None. | None. | Yes | N/A |
| 7. DLC-Based Key Transport | None. | None. | Yes | N/A |
| 8. Key Confirmation | None. | None. | Yes | N/A |
| 8.1 Assurance of Possession Considerations when using Key Confirmation | None. | None. | Yes | N/A |
| 8.2 Unilateral Key Confirmation for Key Agreement Schemes | None. | None. | Yes | N/A |
| 8.3 Bilateral Key Confirmation for Key Agreement Schemes | None. | None. | Yes | N/A |
| 8.4 Incorporating Key Confirmation into a Key Agreement Scheme | None. | None. | Yes | N/A |
| 8.4.1 C(2, 2) Scheme with Unilateral Key Confirmation Provided by U to V | None. | None. | Yes | N/A |
| 8.4.2 C(2, 2) Scheme with Unilateral Key Confirmation | None. | None. | Yes | N/A |

| Section | Exceptions to Shall/Shall Not Statement(s) | Should (Not) Statements[4] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| Provided by V to U | | | | |
| 8.4.3 C(2, 2) Scheme with Bilateral Key Confirmation | None. | None. | Yes | N/A |
| 8.4.4 C(1, 2) Scheme with Unilateral Key Confirmation Provided by U to V | None. | None. | Yes | N/A |
| 8.4.5 C(1, 2) Scheme with Unilateral Key Confirmation Provided by V to U | None. | None. | Yes | N/A |
| 8.4.6 C(1, 2) Scheme with Bilateral Key Confirmation | None. | None. | Yes | N/A |
| 8.4.7 C(1, 1) Scheme with Unilateral Key Confirmation Provided by V to U | None. | None. | Yes | N/A |
| 8.4.8 C(0, 2) Scheme with Unilateral Key Confirmation Provided by U to V | None. | None. | Yes | N/A |
| 8.4.9 C(0, 2) Scheme with Unilateral Key Confirmation Provided by V to U | None. | None. | Yes | N/A |
| 8.4.10 C(0, 2) Scheme with Bilateral Key Confirmation | None. | None. | Yes | N/A |

## 8.4 NIST Special Publication 800-56B

The TOE is compliant with NIST SP 800-56B as described in Table 28 below.

**Table 28 800-56B Compliance**

| Section | Shall/Shall Not Statement(s) | Should (Not) Statements[5] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| 5 Cryptographic Elements | None. | None. | Yes | N/A |
| 5.1 Cryptographic Hash Functions | None. | None. | Yes | N/A |
| 5.2 Message Authentication Code (MAC) Algorithm | None. | None. | Yes | N/A |
| 5.2.1 MacTag Computation | None. | None. | Yes | N/A |
| 5.2.2 MacTag Checking | N/A, no shall statements | None. | Yes | N/A |
| 5.2.3 Implementation Validation Message | None. | None. | Yes | N/A |
| 5.3 Random Bit Generation | None. | None. | Yes | N/A |
| 5.4 Prime Number Generators | Only approved prime number generation methods shall be employed in this Recommendation. | None. | No | TOE is ANSI X9.31 compliant. |
| 5.5 Primality Testing Methods | None. | None. | Yes | N/A |
| 5.6 Nonces | None. | "When using a nonce, a random nonce **should** be used." | Yes | N/A |
| 5.7 Symmetric Key-Wrapping Algorithms | N/A for TLS and SSH. | None. | Yes | N/A |
| 5.8 Mask Generation Function (MGF) | None. | None. | Yes | N/A |
| 5.9 Key Derivation Functions for Key Establishment | None. | None. | Yes | TOE uses other allowable methods and the protocols as |

[5] This column does not included "should/should not" statements that relate to the "owner", "recipient", "application", or "party" as they are outside of the scope of the TOE.

| Section | Shall/Shall Not Statement(s) | Should (Not) Statements[5] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| Schemes | | | | referenced in FIPS 140-2 Annex D |
| 5.9.1 Concatenation Key Derivation Function (Approved Alternative 1) | None. | None. | Yes | N/A |
| 5.9.2 ASN.1 Key Derivation Function (Approved Alternative 2) | None. | None. | Yes | N/A |
| 6 RSA Key Pairs | N/A, no shall statements | None. | Yes | N/A |
| 6.1 General Requirements | None. | "a key pair used for schemes specified in this recommendation **should not** be used for any schemes not specified herein" | Yes | N/A |
| 6.2 Criteria for RSA Key Pairs for Key Establishment | N/A, no shall statements | None. | Yes | N/A |
| 6.2.1 Definition of a Key Pair | None. | None. | Yes | N/A |
| 6.2.2 Formats | N/A, no shall statements | None. | Yes | N/A |
| 6.2.3 Parameter Length Sets | None. | "The MacKey length shall meet or exceed the target security strength, and **should** meet or exceed the security strength of the modulus." | Yes | N/A |
| 6.3 RSA Key Pair Generators | None. | None. | Yes | N/A |
| 6.3.1 RSAKPG1 Family: RSA Key Pair Generation with a Fixed Public Exponent | No shall statements (def of approved key pair generator) | None. | Yes | N/A |
| 6.3.2 RSAKPG2 Family: RSA Key Pair Generation with a Random Public | No shall statements (def of approved key pair generator) | None. | Yes | N/A |

| Section | Shall/Shall Not Statement(s) | Should (Not) Statements[5] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| Exponent | | | | |
| 6.4 Assurances of Validity | N/A, no shall statements | None. | Yes | N/A |
| 6.4.1 Assurance of Key Pair Validity | None. | None. | Yes | N/A |
| 6.4.2 Recipient Assurances of Public Key Validity | None. | None. | Yes | N/A |
| 6.5 Assurances of Private Key Possession | None. | None. | Yes | N/A |
| 6.5.1 Owner Assurance of Private Key Possession | None. | None. | Yes | N/A |
| 6.5.2 Recipient Assurance of Owner's Possession of a Private Key | None. | None. | Yes | N/A |
| 6.6 Key Confirmation | None. | None. | Yes | N/A |
| 6.6.1 Unilateral Key Confirmation for Key Establishment Schemes | Unilateral Key Confirmation is done for both TLS and SSH, however it varies slightly from that outlined here. | None. | Yes | N/A |
| 6.6.2 Bilateral Key Confirmation for Key Establishment Schemes | N/A, no shall statements | None. | Yes | N/A |
| 6.7 Authentication | N/A, no shall statements | None. | Yes | N/A |
| 7 IFC Primitives and Operations | N/A, no shall statements | None. | Yes | N/A |
| 7.1 Encryption and Decryption Primitives | N/A, no shall statements | None. | Yes | N/A |
| 7.1.1 RSAEP | N/A, no shall statements | None. | Yes | N/A |

| Section | Shall/Shall Not Statement(s) | Should (Not) Statements[5] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| 7.1.2 RSADP | N/A, no shall statements | "Care **should** be taken to ensure that an implementation of RSADP does not reveal even partial information about the value of k." | Yes | N/A |
| 7.2 Encryption and Decryption Operations | N/A, no shall statements | None. | Yes | N/A |
| 7.2.1 RSA Secret Value Encapsulation (RSASVE) | N/A, no shall statements | "Care **should** be taken to ensure that an implementation does not reveal information about the encapsulated secret value Z."<br>"the observable behavior of the I2BS routine should not reveal even partial information about the byte string Z." | Yes | N/A |
| 7.2.2 RSA with Optimal Asymmetric Encryption Padding (RSA-OAEP) | None. | "Care should be taken to ensure that the different error conditions that may be detected in Step 5 above cannot be distinguished from one another by an opponent, whether by error message or by process timing."<br>"A single error message **should** be employed and output the same way for each type of decryption error. There **should** be no difference in the observable behavior for the different RSA-OAEP decryption errors."<br>"care should be taken to ensure that even if there are no errors, an implementation does not reveal partial information about the encoded message EM"<br>"the observable behavior of the mask generation function **should not** reveal even partial information about the MGF seed employed in the | Yes | N/A |

| Section | Shall/Shall Not Statement(s) | Should (Not) Statements[5] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| | | process" | | |
| 7.2.3 RSA-based Key-Encapsulation Mechanism with a Key-Wrapping Scheme (RSA-KEM-KWS) | N/A, no shall statements | "Care **should** be taken to ensure that the different error conditions in Steps 2.2, 4, and 6 cannot be distinguished from one another by an opponent, whether by error message or timing."<br>"A single error message **should** be employed and output the same way for each error type. There **should** be no difference in timing or other behavior for the different errors. In addition, care **should** be taken to ensure that even if there are no errors, an implementation does not reveal partial information about the shared secret<br>Z."<br>"care **should** be taken to ensure that an implementation does not reveal information about the encapsulated secret value Z. For instance, the observable behavior of the KDF **should not** reveal even partial information about the Z value employed in the key derivation process." | Yes | N/A |
| 8 Key Agreement Schemes | In many cases TLS is deployed only with server authentication. | None. | Yes | N/A |
| 8.1 Common Components for Key Agreement | N/A, no shall statements | None. | Yes | N/A |
| 8.2 The KAS1 Family | N/A, no shall statements | None. | Yes | N/A |
| 8.2.1 KAS1 Family Prerequisites | None. | None. | Yes | N/A |
| 8.2.2 KAS1-basic | None. | None. | Yes | N/A |

| Section | Shall/Shall Not Statement(s) | Should (Not) Statements[5] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| 8.2.3 KAS1 Key Confirmation | None. | None. | Yes | N/A |
| 8.2.4 KAS1 Security Properties | N/A, no shall statements | None. | Yes | N/A |
| 8.3 The KAS2 Family | N/A, no shall statements | None. | Yes | N/A |
| 8.3.1 KAS2 Family Prerequisites | None. | None. | Yes | N/A |
| 8.3.2 KAS2-basic | None. | "the observable behavior of the key-agreement process **should not** reveal partial information about the shared secret Z." | Yes | N/A |
| 8.3.3 KAS2 Key Confirmation | None. | None. | Yes | N/A |
| 8.3.4 KAS2 Security Properties | N/A, no shall statements | None. | Yes | N/A |
| 9 IFC based Key Transport Schemes | None. | None. | Yes | N/A |
| 9.1 Additional Input | None. | None. | Yes | N/A |
| 9.2 KTS-OAEP Family: Key Transport Using RSA-OAEP | N/A, no shall statements | None. | Yes | N/A |
| 9.2.1 KTS-OAEP Family Prerequisites | None. | None. | Yes | N/A |
| 9.2.2 Common components | N/A, no shall statements | None. | Yes | N/A |
| 9.2.3 KTS-OAEP-basic | None. | None. | Yes | N/A |
| 9.2.4 KTS-OAEP Key Confirmation | None. | None. | Yes | N/A |
| 9.2.5 KTS-OAEP Security Properties | N/A, no shall statements | None. | Yes | N/A |
| 9.3 KTS-KEM-KWS Family: Key Transport | N/A, no shall | None. | Yes | N/A |

| Section | Shall/Shall Not Statement(s) | Should (Not) Statements[5] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| using RSA-KEM-KWS | statements | | | |
| 9.3.1 KTS-KEM-KWS Family Prerequisites | None. | None. | Yes | N/A |
| 9.3.2 Common Components of the KTS-KEM-KWS Schemes | N/A, no shall statements | None. | Yes | N/A |
| 9.3.3 KTS-KEM-KWS-basic | None. | None. | Yes | N/A |
| 9.3.4 KTS-KEM-KWS Key Confirmation | None. | None. | Yes | N/A |
| 9.3.5 KTS-KEM-KWS Security Properties | N/A, no shall statements | None. | Yes | N/A |

## 8.5 FIPS PUB 186-3, Appendix B Compliance

The TOE is compliant as described in the table below.

**Table 29 FIPS PUB 186-3, Appendix B Compliance**

| Section | Exceptions to Shall/Shall Not Statement(s) | Exceptions to Should Statements[6] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| B.1 FFC Key Pair Generation | Not Implemented. | N/A | N/A | FFC Key Pair Generation Not Implemented |
| B.1.1 Key Pair Generation Using Extra Random Bits | Not Implemented. | N/A | N/A | Not Implemented. |

[6] This column does not included “should/should not” statements that relate to the “owner”, “recipient”, “application”, or “party” as they are outside of the scope of the TOE.

| Section | Exceptions to Shall/Shall Not Statement(s) | Exceptions to Should Statements[6] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| B.1.2 Key Pair Generation by Testing Candidates | Not Implemented. | N/A | N/A | Not Implemented. |
| B.2 FFC Per-Message Secret Number Generation | Not Implemented. | N/A | N/A | Not Implemented. |
| B.2.1 Per-Message Secret Number Generation Using Extra Random Bits | Not Implemented. | N/A | N/A | Not Implemented. |
| B.2.2 Per-Message Secret Number Generation by Testing Candidates | Not Implemented. | N/A | N/A | Not Implemented. |
| B.3 IFC Key Pair Generation | N/A | N/A | Yes | N/A |
| B.3.1 Criteria for IFC Key Pairs | None. | N/A | Yes | N/A |
| B.3.2 Generation of Random Primes that are Provably Prime | Not Implemented. | N/A | N/A | TOE does not implement this prime generation method, but does the method in Section B.3.4. |
| B.3.2.1 Get the Seed | None. | N/A | Yes | N/A |
| B.3.2.2 Construction of the Provable Primes p and q | Not Implemented. | N/A | N/A | TOE does not implement this prime generation method, but does the method in Section B.3.4. |
| B.3.3 Generation of Random Primes that are Probably Prime | Not Implemented. | N/A | N/A | TOE does not implement this prime generation method, but does the method in Section B.3.4. |
| B.3.4 Generation of Provable Primes with Conditions Based on Auxiliary Provable Primes | None. | N/A | Yes | N/A |

| Section | Exceptions to Shall/Shall Not Statement(s) | Exceptions to Should Statements[6] | TOE Compliant? | Rationale |
|---|---|---|---|---|
| B.3.5 Generation of Probable Primes with Conditions Based on Auxiliary Provable Primes | Not Implemented. | N/A | N/A | TOE does not implement this prime generation method, but does the method in Section B.3.4. |
| B.3.6 Generation of Probable Primes with Conditions Based on Auxiliary Probable Primes | Not Implemented. | N/A | N/A | TOE does not implement this prime generation method, but does the method in Section B.3.4. |
| B.4 ECC Key Pair Generation | None. | N/A | Yes | N/A |
| B.4.1 Key Pair Generation Using Extra Random Bits | None. | On error, invalid values for d and Q are not returned; instead, no key at all is returned. | Yes | The structure of the code doesn't return values for d and Q; instead, on success, the generated keys are installed. |
| B.4.2 Key Pair Generation by Testing Candidates | Not Implemented. | None. | N/A | TOE does not implement this prime generation method. |
| B.5 ECC Per-Message Secret Number Generation | None. | N/A | Yes | N/A |
| B.5.1 Per-Message Secret Number Generation Using Extra Random Bits | None. | On error, invalid values of k and k^-1 are not returned. | Yes | On error, k and k^-1 aren't used. |
| B.5.2 Per-Message Secret Number Generation by Testing Candidates | Not Implemented. | None. | N/A | TOE does not implement this method of ECC Signature Generation. |

# 9 ANNEX A: REFERENCES

The following documentation was used to prepare this ST:

**Table 30: References**

| Identifier | Description |
|---|---|
| [CC_PART1] | Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-001 |
| [CC_PART2] | Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-002 |
| [CC_PART3] | Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-003 |
| [CEM] | Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-004 |
| [NDPP]<br>[TFFWEP]<br>[VPNGWEP] | U.S. Government Protection Profiles for Security Requirements for Network Devices (NDPP), Traffic Filter Firewall Extended Package (TFFWEP), and VPN Gateway Extended Package (VPNGWEP) |
| [800-38A] | NIST Special Publication 800-38A Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001 |
| [800-56A] | NIST Special Publication 800-56A, March, 2007<br>Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography (Revised) |
| [800-56B] | NIST Special Publication 800-56B Recommendation for Pair-Wise, August 2009<br>Key Establishment Schemes Using Integer Factorization Cryptography |
| [FIPS 140-2] | FIPS PUB 140-2 Federal Information Processing Standards Publication<br>Security Requirements for Cryptographic Modules May 25, 2001 |
| [FIPS PUB 186-2] | FIPS PUB 186-2 Federal Information Processing Standards Publication 2000 January 27 |
| [FIPS PUB 186-3] | FIPS PUB 186-3 Federal Information Processing Standards Publication Digital Signature Standard (DSS) June 2009 |
| [FIPS PUB 186-4] | FIPS PUB 186-3 Federal Information Processing Standards Publication Digital Signature Standard (DSS) July 2013 |
| [FIPS PUB 198-1] | Federal Information Processing Standards Publication The Keyed-Hash Message Authentication Code (HMAC) July 2008 |
| [800-90] | NIST Special Publication 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators January 2012 |
| [FIPS PUB 180-3] | FIPS PUB 180-3 Federal Information Processing Standards Publication Secure Hash Standard (SHS) October 2008 |
| [FIPS PUB 180-4] | FIPS PUB 180-4 Federal Information Processing Standards Publication Secure Hash Standard (SHS) March 2012 |