Aruba Networks

Virtual Mobility Controller (hardened Chassis running VMware ESXi) with ArubaOS 6.4.2.0-1.3 FIPS

NDPP/TFFW-EP/VPNGW-EP

Security Target

April 2017

aruba

a Hewlett Packard Enterprise company

Document prepared by

CSC

BUSINESS SOLUTIONS
TECHNOLOGY
OUTSOURCING

Document History

Version	Date	Author	Description
0.1	June 10, 2015	B Pleffner	Initial Draft
0.2	December 4, 2015	B Pleffner	Updates per lab comments
0.3	January 18, 2016	B Pleffner	Updates per ASD comments
0.4	April 13, 2016	B Pleffner	Update TOE version number
0.5	March 3, 2017	S. Weingart	Update tested platforms and company logo and details
1.0	April 26, 2017	B. Pleffner	Final Release

Table of Contents

List of Tables

List of Figures

1 Introduction

1.1 Overview

1 The Aruba Networks Virtual Mobility Controller (VMC) is a virtualized network device that serves as a gateway between wired and wireless networks and provides command-and-control over Access Points (APs) within an Aruba dependant wireless network. ArubaOS 6.4.2.0-1.3 FIPS is the underlying operating system of the VMC, which runs on top of VMware ESXi and was evaluated on the following hardware platforms:

a) PacStar 451 Small Server Module (Intel 4th-Generation Core i5 or Core i7)

b) Information Assurance Specialists IAS Router MICRO Extreme network appliance (contains the IAS VPN Gateway Module CLASSIC using Intel 4th-Generation Core i5)

c) Klas Telecom Voyager VMm (Intel 5th-Generation Core i3)

d) DTECH Labs M3-SE-SVR3Q (Intel 3rd-Generation Core i7)

2 This Security Target (ST) defines the Virtual Mobility Controller (hardened Chassis running VMware ESXi) with ArubaOS 6.4.2.0-1.3 FIPS Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

3 Whilst the Aruba Networks VMC offers a wide range of wireless, wired and remote networking features, the TOE is constrained to the following security features:

a) Secure communication with remote administrators, authentication servers and audit servers

b) Secure management including authentication, verifiable updates and auditing

c) Self verification of integrity and operation

d) Stateful traffic filter firewall

VPN gateway functionality

4 For a precise statement of the scope of incorporated security features, refer to section 2.4.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	Aruba Networks Virtual Mobility Controller (hardened Chassis running VMware ESXi) with ArubaOS 6.4.2.0-1.3 FIPS Software Version: 6.4.2.0-1.3 FIPS
Security Target	Aruba Networks Virtual Mobility Controller (hardened Chassis running VMware ESXi) with ArubaOS 6.4.2.0-1.3 FIPS NDPP/TFFW-EP/VPNGW-EP Security Target, v1.0

1.3 Conformance Claims

5 This ST supports the following conformance claims:

a) CC version 3.1 release 3

b) CC Part 2 extended

c) CC Part 3 conformant

d) U.S. Government Approved Protection Profile - Security Requirements for Network Devices , v1.1 with Errata #3 applied (herein referred to as NDPP)

e) U.S. Government Approved Protection Profile - Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Filter Firewall, v1.0 (herein referred to as TFFW-EP)

f) U.S. Government Approved Protection Profile - Network Device Protection Profile (NDPP) Extended Package VPN Gateway, v1.1 (herein referred to as VPNGW-EP)

1.4 Terminology

Table 2: Terminology

Term	Definition
ACL	Access Control List
AP	Access Point
ARM	Adaptive Radio Management
CC	Common Criteria
CLI	Command Line Interface
CP	Control Plane
CSP	Critical Security Parameter
DP	Data Plane
EAL	Evaluation Assurance Level
GUI	Graphical User Interface
IKE	Internet Key Exchange
KAT	Known Answer Test
LDAP	Lightweight Directory Access Protocol
NDPP	U.S. Government Approved Protection Profile – Security Requirements for Network Devices , v1.1 with Errata #e applied
NIST	National Institute of Standards and Technology
NTP	Network Time Protocol
OSP	Organizational Security Policy

Term	Definition
PP	Protection Profile
RAP	Remote Access Point
RF	Radio Frequency
ST	Security Target
TFFW-EP	U.S. Government Approved Protection Profile – Network Device Protection Profile (NDPP) Extended Package Stateful Traffic Filter Firewall, v1.0
TOE	Target of Evaluation
TSF	TOE Security Functionality
VMC	Virtual Mobility Controller
VPN	Virtual Private Network
VPNGW-EP	U.S. Government Approved Protection Profile - Network Device Protection Profile (NDPP) Extended Package VPN Gateway, v1.1
WebUI	Web User Interface

1.5 References

[USER]	ArubaOS 6.4.2 User Guide, Ref 0511615-00v1
[CLI]	ArubaOS 6.4.2 Command Line Interface, Ref 0511616-00v1
[SYSLOG]	ArubaOS 6.4. Syslog Messages Guide, Ref 0511324-02
[MIB]	ArubaOS 6.x MIB Reference Guide, Ref 0511323-02
[FIPS]	Aruba 600/3000/6000/7200 FIPS 140-2 Security Policy

2 TOE Description

2.1 Type

6 The TOE is a network device, stateful traffic filter firewall and VPN gateway.

7 In the CC evaluated configuration, the TOE must be configured to operate in the FIPS 140-2 Approved mode of operation. In FIPS-Approved mode, various weak protocols and algorithms are disabled. Please reference the appropriate FIPS 140-2 Security Policy documents for each controller and access point for more details at http://csrc.nist.gov/groups/STM/cmvp/index.html.

2.2 TOE Architecture

8 At a high level, the Aruba VMC is a 64-bit virtualized software-based WLAN, VPN, and firewall solution on x86 architecture. The VMC is the first hop for data traffic on virtualized network infrastructure. The VMC operates on x86 platforms in VMware environment and can reside with other virtualized appliances. The software running on the VMC is called ArubaOS, which consists of two main components:

a) Control Plane (CP)—implements functions which can be handled at lower speeds such as VMC system management (CLI and Web GUI), user authentication (e.g. 802.1X, RADIUS, LDAP), Internet Key Exchange (IKE), auditing/logging (syslog), Wireless IDS (WIDS), and termination of protocols operating at the system level (e.g. SSH, TLS, NTP, etc.). The Control Plane runs the Linux operating system along with various user-space applications (described below).

b) Data Plane (DP)—implements functions that must be handled at high speeds such as high-speed switching functions (forwarding, VLAN tagging/enforcement, bridging), termination of 802.11 associations/sessions, tunnel termination (GRE, IPsec), stateful firewall and deep packet inspection functions, and cryptographic acceleration. In the VMC, the Data Plane is implemented as a Linux process that makes use of the Intel Data Plane Development Kit (DPDK) framework.

9 The Control Plane and Data Plane are inseparable. Administrators install the virtual machine by creating a virtual machine in the ESXi client interface and then applying the VMC OVF template to the virtual machine, identified as “ArubaOS”. Internally, the VMC unpacks the ArubaOS software image into its various components. A given ArubaOS software image has a single version number, and includes all software components necessary to operate both VMCs and APs.

10 The CP runs the Linux OS, along with various custom user-space applications which provide the following CP functions:

a) Monitors and manages critical system resources, including processes, memory, and flash

b) Manages system configuration and licensing

c) Manages an internal database used to store licenses, user authentication information, etc

d) Provides network anomaly detection, hardware monitoring, mobility management, wireless management, and radio frequency management services

e) Provides a Command Line Interface (CLI)

f) Provides a web-based (HTTPS/TLS) management UI for the VMC

g) Provides various WLAN station and AP management functions

h) Provides authentication services for the system management interfaces (CLI, web GUI) as well as for WLAN users

i) Provides IPsec key management services for APs, VPN users, and connections with other Aruba Mobility Controllers (**Note:** IPsec for APs, VPN users and other Mobility Controllers or Virtual Mobility Controllers are not within the scope of evaluation)

j) Provides network time protocol service for APs, point to point tunneling protocol services for users, layer 2 tunneling protocol services for users, SSH services for incoming management connections, SNMP client/agent services, and protocol independent multicast (routing) services for the controller

k) Provides syslog services by sending logs to the operating environment.

11 The Linux OS running on the CP is a version 2.6.32 kernel. Linux is a soft real-time, multi-threaded operating system that supports memory protection between processes. Only Aruba provided interfaces are used, and the CLI is a restricted command set. Administrators do not have access to the Linux command shell or operating system.

2.3 Usage

12 The TOE is generally deployed as a gateway between wired and wireless networks that performs command-and-control within an Aruba dependent wireless network architecture consisting of one or more Aruba VMCs and multiple Aruba wireless APs. In this architecture, Aruba split the traditional functions of an all-in-one wireless access point between the controller and the AP. A simple TOE deployment is depicted in Figure 1.

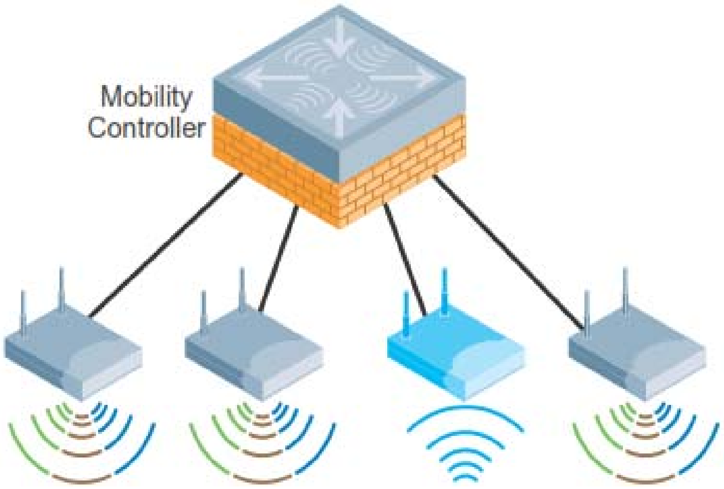

Figure 1: TOE usage scenario

13 There are many combinations of deployment scenarios, ranging from branch office environments in which the VMC and access point are combined to campus deployments with multiple redundant VMCs.

14 The non-security functionality provided by a VMC goes beyond managing dependants APs, and includes:

a) Performing Layer 2 switching and Layer 3 routing

b) Terminating Internet-based remote access points

c) Providing advanced Radio Frequency (RF) services with Adaptive Radio Management (ARM) and spectrum analysis

d) Providing location services and RF coverage “heat maps” of the deployment

e) Providing self-contained management by way of a master/local hierarchy with one controller

f) Pushing configuration to other VMCs to reduce administrative overhead

g) Delivering AP software updates automatically when the VMC is upgraded

2.4 Security Functions

15 The TOE provides the following security functions:

a) **Protected communications.** The TOE protects the following communication flows:

i) **WebUI.** Communication with the administrative web user interface (WebUI) is protected using TLS/HTTPS.

ii) **CLI.** Remote administration via the Command Line Interface (CLI) is protected using SSHv2.

iii) **Syslog.** Syslog messages are protected using IPSec.

iv) **Radius.** Radius authentication messages are protected using IPSec.

b) **Verifiable updates.** Updates are digitally signed and verified upon installation utilizing digital signatures.

c) **System monitoring.** The TOE maintains an audit log of administrative and security relevant events. Logs can optionally be delivered to a Syslog server.

d) **Secure administration.** The TOE provides administration interfaces for configuration and monitoring. The TOE authenticates administrators and implements session timeouts.

e) **Residual information clearing.** The TOE ensures that network packets sent from the TOE do not include data "left over" from the processing of previous network information.

f) **Self-test.** The TOE performs both power-up and conditional self-tests to verify correct and secure operation.

g) **Firewall.** The TOE performs stateful packet filtering. Wireless clients connecting through APs are placed into user-roles. Stateful packet filter policies are applied to these user-roles to allow fine grained control over wireless traffic.

h) **VPN gateway.** The TOE may be used as a VPN gateway – a device at the edge of a private network that terminates an IPsec tunnel, which provides device authentication, confidentiality, and integrity of information traversing a public or untrusted network.

2.5 Physical Scope

16 The TOE comprises the ArubaOS 6.4.2.0-1.3 FIPS software, VMware ESXi, and any hardware platform with an Intel x86-64 CPU that supports the Intel RDRAND, AES-NI, and VT-x instruction sets. The evaluated configuration included:

a) VMware ESXi 5.5;

b) PacStar 451 Small Server Module containing an Intel 4th-Generation quad-core Core i7 CPU with 16GB RAM;

c) Information Assurance Specialists IAS Router MICRO Extreme network appliance containing the “IAS VPN Gateway Module CLASSIC” using an Intel 4th-Generation Core i5 with 16GB RAM;

d) Klas Telecom Voyager VMm containing an Intel 5th-Generation Core i3 with 32GB RAM;

e) DTECH Labs M3-SE-SVR3 containing an Intel 3rd-Generation Core i7 with 16GB RAM.

17 ArubaOS 6.4.2.0-1.3 FIPS consists of a base software package with add-on software modules that can be activated by installing the appropriate licenses. The following licenses are required for the evaluated configuration (and are within the physical scope):

a) Advanced Cryptography **Note:** Only required if using Elliptic Curve cryptography or AES-GCM

b) Policy Enforcement Firewall Next Generation

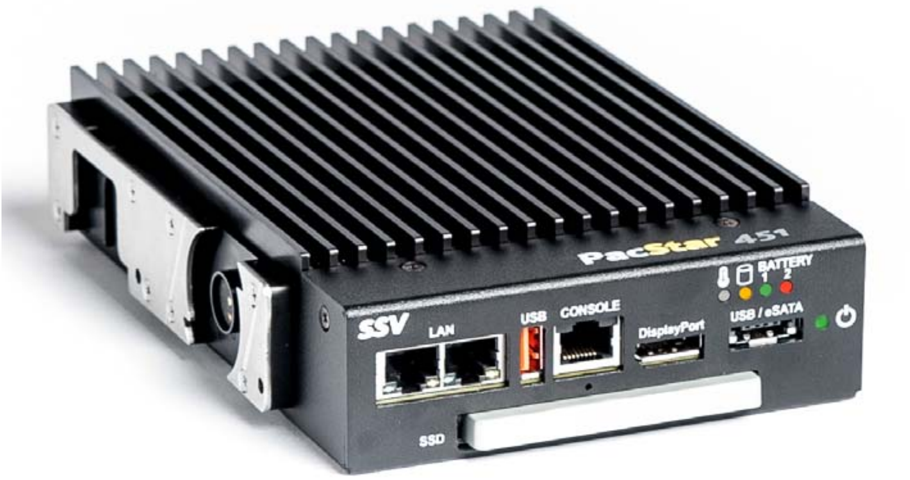

Figure 2: PacStar 451 SSV Chassis

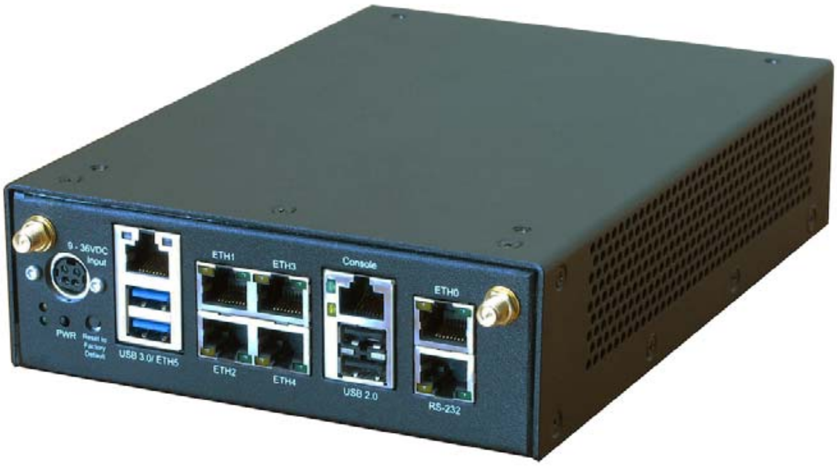

Figure 3: IAS Router MICRO Extreme

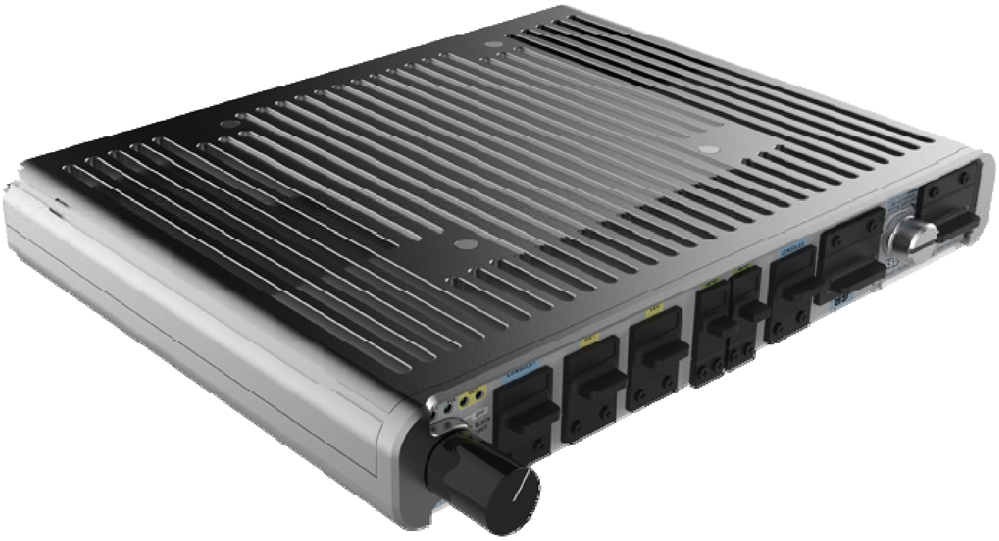

Figure 4: Klas Telecom Voyager VMm

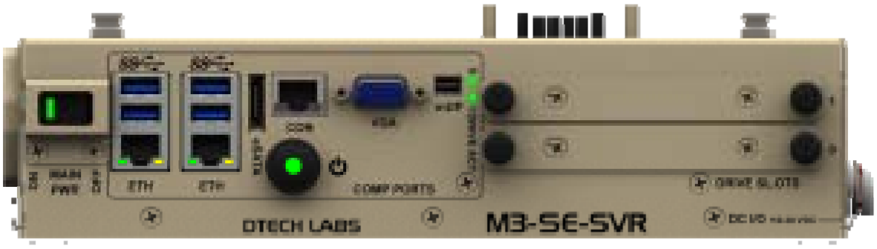

Figure 5: DTECH M3-SE-SVR

2.5.1 Guidance Documents

18 The TOE includes the following guidance documents:

a) ArubaOS 6.4.2 User Guide, Ref 0511615-00v1

b) ArubaOS 6.4.2 Command Line Interface, Ref 0511616-00v1

c) ArubaOS 6.4. Syslog Messages Guide, Ref 0511324-02

d) ArubaOS 6.x MIB Reference Guide, Ref 0511323-02

e) Aruba 600/3000/6000/7200 FIPS 140-2 Security Policy

2.5.2 Non-TOE Components

19 The TOE operates with the following components in the environment:

a) **Access Points.** APs connect to the TOE in Aruba dependent wireless network architectures. Wireless clients connect to the APs.

b) **Audit Server.** The TOE can utilize a Syslog server to store audit records.

c) **Authentication Server.** The TOE can utilize a Radius server to authenticate users.

d) **Time Server.** The TOE can utilize a Network Time Protocol (NTP) server to synchronize its system clock with a central time source.

e) **Web Browser.** The remote administrator can use a web browser to access the Web GUI interface.

f) **SSH Client.** The remote administrator can use an SSH client to access the CLI.

g) **VPN Client.** When acting as a VPN gateway, the TOE may communicate with other VPN gateways or with VPN clients. The VPN clients may be hardware devices (e.g. Aruba Remote Access Point) or may be implemented as software (e.g. Aruba VIA Client).

2.6 Logical Scope

20 The logical scope of the TOE comprises the security functions defined in section 2.4.

3 Security Problem Definition

3.1 Threats

Table 3: Threats drawn from NDPP

Identifier	Description
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.
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.

Table 4: Threats drawn from TFFW-EP

Identifier	Description
T.NETWORK_DISCLOSURE	Sensitive information on a protected network might be disclosed resulting from ingress- or egress-based actions.
T. NETWORK_ACCESS	Unauthorized access may be achieved to services on a protected network from outside that network, or alternately services outside a protected network from inside the protected network.
T.NETWORK_MISUSE	Access to services made available by a protected network might be used counter to Operational Environment policies.

Identifier	Description
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.

Table 5: Threats drawn from VPNGW-EP

Identifier	Description
T.NETWORK_DISCLOSURE	Sensitive information on a protected network might be disclosed resulting from ingress- or egress-based actions.
T. NETWORK_ACCESS	Unauthorized access may be achieved to services on a protected network from outside that network, or alternately services outside a protected network from inside the protected network.
T.NETWORK_MISUSE	Access to services made available by a protected network might be used counter to Operational Environment policies.
T.TSF_FAILURE	Security mechanisms of the TOE mail fail, leading to a compromise of the TSF.
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.
T.UNPROTECTED_CLIENT_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.

3.2 Organizational Security Policies

Table 6: OSPs drawn from NDPP

Identifier	Description
P.ACCESS_BANNER	The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE.

3.3 Assumptions

Table 7: Assumptions drawn from NDPP

Identifier	Description
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.

Table 8: Assumptions drawn from TFFW-EP

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

Table 9: Assumptions drawn from VPNGW-EP

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

4 Security Objectives

4.1 Objectives for the Operational Environment

Table 10: Operational environment objectives drawn from NDPP

Identifier	Description
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.

Table 11: Operational environment objectives drawn from TFPP

Identifier	Description
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.

Table 12: Operational environment objectives drawn from VPNPP

Identifier	Description
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.

4.2 Objectives for the TOE

Table 13: Objectives drawn from NDPP

Identifier	Description
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.

Identifier	Description
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.

Table 14: Objectives drawn from TFFW-EP

Identifier	Description
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 network packets based on source and destination transport layer ports.
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.

Table 15: Objectives drawn from VPNGW-EP

Identifier	Description
O.ADDRESS_FILTERING	The TOE will provide the means to filter and log network packets based on source and destination addresses.
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.

Identifier	Description
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.PORT_FILTERING	The TOE will provide the means to filter and log network packets based on source and destination transport layer ports.
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 a 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 the VPN headend assigning an IP address that the gateway controls, as well as providing a routing point for the client's network traffic.

5 Security Requirements

5.1 Conventions

21 This document uses the following font conventions to identify the operations defined by the CC:

a) **Assignment.** Indicated with italicized text.

b) **Refinement.** Indicated with bold text and strikethroughs.

c) **Selection.** Indicated with underlined text.

d) **Assignment within a Selection:** Indicated with italicized and underlined text.

e) **Iteration.** Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).

22 Operations specified by the NDPP (that are not specified by CC Part 2) are also identified using the above convention.

23 Explicitly stated SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs.

24 Application notes from the NDPP have not been reproduced except where their inclusion aids the ST reader in understanding the SFRs.

5.2 Extended Components Definition

25 Table 16 identifies the extended components which are incorporated into this ST. All components are reproduced directly from the NDPP, TFFW-EP and VPNGW-EP - no further definition is provided in this document.

Table 16: Extended Components

Component	Title	Source
FAU_STG_EXT.1	External Audit Trail Storage	NDPP
FCS_CKM_EXT.4	Cryptographic Key Zeroization	NDPP
FCS_RBG_EXT.1	Cryptographic Operation (Random Bit Generation)	NDPP VPNGW-EP
FIA_PMG_EXT.1	Password Management	NDPP
FIA_UIA_EXT.1	User Identification and Authentication	NDPP
FIA_UAU_EXT.2	Password-based Authentication Mechanism	NDPP
FIA_PSK_EXT.1	Extended: Pre-Shared Key Composition	NDPP VPNGW-EP
FIA_X509_EXT.1	Extended: X.509 Certificates	VPNGW-EP
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)	NDPP

Component	Title	Source
FPT_APW_EXT.1	Protection of Administrator Passwords	NDPP
FPT_TUD_EXT.1	Trusted Update	NDPP VPNGW-EP
FPT_TST_EXT.1	TSF Testing	NDPP VPNGW-EP
FTA_SSL_EXT.1	TSF-initiated Session Locking	NDPP
FTA_VCM_EXT.1	VPN Client Management	VPNGW-EP
FCS_IPSEC_EXT.1	Explicit: IPSEC	NDPP VPNGW-EP
FCS_TLS_EXT.1	Explicit: TLS	NDPP
FCS_SSH_EXT.1	Explicit: SSH	NDPP
FFW_RUL_EXT.1	Stateful Traffic Filtering	TFFW-EP
FPF_RUL_EXT.1	Packet Filtering	VPNGW-EP

5.3 Functional Requirements

Table 17: Summary of SFRs

Requirement	Title
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	External Audit Trail Storage
FCS_CKM.1(1)	Cryptographic Key Generation (for asymmetric keys – HTTPS/TLS)
FCS_CKM.1(2)	Cryptographic Key Generation (for asymmetric keys – IPSec)
FCS_CKM.1(3)	Cryptographic Key Generation (for asymmetric keys – SSH)
FCS_CKM.1(4)	Cryptographic Key Generation (for asymmetric keys – IPSec/IKE)
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 – RSA)
FCS_COP.1(3)	Cryptographic Operation (for cryptographic hashing)

Requirement	**Title**
FCS_COP.1(4)	Cryptographic Operation (for cryptographic signature - ECDSA)
FCS_COP.1(5)	Cryptographic Operation (for cryptographic signature – ECDSA)
FCS_RBG_EXT.1(1)	Extended: Cryptographic Operation (Random Bit Generation – SSH/TLS)
FCS_RBG_EXT.1(2)	Extended: Cryptographic Operation (Random Bit Generation - IPSec)
FCS_HTTPS_EXT.1	Explicit: HTTPS
FCS_TLS_EXT.1	Explicit: TLS
FCS_IPSEC_EXT.1	Explicit: IPSEC
FCS_SSH_EXT.1	Explicit: SSH
FDP_RIP.2	Full Residual Information Protection
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
FIA_AFL.1	Authentication Failure Handling
FIA_PSK_EXT.1	Extended: Pre-Shared Key Composition
FIA_X509_EXT.1	Extended: X.509 Certificates
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
FMT_MOF.1	Management of Security Functions Behavior
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	TSF Testing

Requirement	Title
FPT_FLS.1	Fail Secure
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3 (1)	TSF-initiated Termination (NDPP)
FTA_SSL.3 (2)	TSF-initiated Termination (VPNGW-EP)
FTA_SSL.4	User-initiated Termination
FTA_TAB.1	Default TOE Access Banners
FTA_TSE.1	TOE Session Establishment
FTA_VCM_EXT.1	VPN Client Management
FTP_ITC.1	Inter-TSF trusted channel
FTP_TRP.1	Trusted Path
FFW_RUL_EXT.1	Stateful Traffic Filtering
FPF_RUL_EXT.1	Packet Filtering

5.3.1 Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

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

a) Start-up and shut-down 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 18*.

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 18*.

Table 18: Auditable events

Requirement	Auditable Events	Additional Audit Record Contents	Guidance Notes

Requirement	Auditable Events	Additional Audit Record Contents	Guidance Notes
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt	See [SYSLOG] for messages regarding I & A, these include: 501199, 541003 , 5220008, 522010, 500008, 5000334, 522017, 522021, 522022
FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).	Same audit messages apply as for FIA_UIA_EXT.1. 501199 specifically records the user and IP address.
FIA_X509_EXT.1	Establishing session with CA	Source and destination addresses Source and destination ports TOE Interface	Audit messages for these actions are stored in the configuration audit trail. For identification, all certificate management commands will include the keywords "crypto-local pki" with the rest of the message indicating whether a certificate was loaded or removed.
FPT_STM.1	Changes to the time. The old and new values for the time.	Origin of the attempt (e.g., IP address).	See [SYSLOG] The time is set by the logged in admin, the log message is" SYSTEM: clock changed from [old_timestamp] to [new_timestamp]" The login authentication message for the Admin will show the IP address (message 501199)
FPT_TUD_EXT.1	Initiation of update.	No additional information.	The audit trail will indicate when a new software image has been copied to the TOE through use of the "copy" command. A complete reboot is required to make an update actually take effect. Message 307404 indicates the process is flagged to start.
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	No additional information.	N/A for this TOE. Interactive sessions are only terminated, not locked.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	No additional information.	See [SYSLOG] – Messages 103042,307354, 203046 (normal termination messages as we do not do locking)
FTA_SSL.4	The termination of an interactive session.	No additional information.	See [SYSLOG] – Messages: 500078, 501198
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.	The Inter-TSF trusted channel is IPsec. Audit messages will be the same as for FCS_IPSEC_EXT.1.

Requirement	Auditable Events	Additional Audit Record Contents	Guidance Notes
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.	Depending on whether the remote administrator is using HTTPS or SSH, the audit messages will be the same as FCS_SSH_EXT.1 or FCS_HTTPS_EXT.1. Audit message 125022 includes the identification of the claimed user identity.
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure.	See [SYSLOG] message ID 103001 through 103092
	Establishment/Termination of an IPsec SA.	Non-TOE endpoint of connection (IP address) for both successes and failures.	See [SYSLOG] message ID 103009, 103077
	Session Establishment with peer **Application Note:** For session establishment, the expectation is that the TOE is capable of auditing all of the packets associated with the establishment of a session; this would include the IKE phase 1 and phase 2 negotiations. The TOE must be able to log all of the packets in a successful session establishment, and also have the ability to log any packets that were dropped or discarded.	Source and destination addresses Source and destination ports TOE Interface	See [SYSLOG] message ID 103009 - 103092
FCS_TLS_EXT.1	Failure to establish a TLS Session.	Reason for failure.	TLS is only used in the context of HTTPS. Audit messages for TLS will be the same as FCS_HTTPS_EXT.1.
	Establishment/Termination of a TLS session.	Non-TOE endpoint of connection (IP address) for both successes and failures.	TLS is only used in the context of HTTPS. Audit messages for TLS will be the same as FCS_HTTPS_EXT.1.
FCS_SSH_EXT.1	Failure to establish an SSH session	Reason for failure.	See [SYSLOG] message ID 125022
	Establishment/Termination of an SSH session	Non-TOE endpoint of connection (IP address) for both successes and failures.	See [SYSLOG] message ID 125032, 125033 (establishment success & Failure). Also see [SYSLOG] Table 16, page 194 for CLI user logging
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure.	See [SYSLOG] message ID 125022
	Establishment/Termination of a HTTPS session.	Non-TOE endpoint of connection (IP address) for both successes and failures.	See [SYSLOG] see Table 16, page 194 for HTTP/S webui user logging

Requirement	Auditable Events	Additional Audit Record Contents	Guidance Notes
FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface	See [SYSLOG] message ID 124006
	Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets.	See [SYSLOG] message ID 304017
FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	Source and destination addresses Source and destination ports Transport Layer Protocol TOE Interface	See [SYSLOG] message ID 124006
	Indication of packets dropped due to too much network traffic	TOE interface that is unable to process packets	See [SYSLOG] message ID 304017

FAU_GEN.2 User Identity Association

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

FAU_STG_EXT.1 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)

FCS_CKM.1(1) Cryptographic Key Generation (for asymmetric keys – HTTPS/TLS)

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*.

Application Note: This requirement is related to the use of RSA in HTTPS/TLS.

FCS_CKM.1(2) Cryptographic Key Generation (for asymmetric keys – IPSec)

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 finite field-based key establishment schemes;*
- *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”)* ***and***
- *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 requirement is related to the use of Diffie-Hellman, RSA and/or ECDSA in IPSec (depending on configuration for the multiple uses of IPSec).

Application Note: This SFR instantiates the FCS_CKM.1 from NDPP and FCS_CKM.1(1) from VPNGW-EP.

FCS_CKM.1(3) Cryptographic Key Generation (for asymmetric keys – SSH)

FCS_CKM.1.1(3) **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 finite field-based key establishment schemes;*

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

Application Note: This requirement is related to the use of Diffie-Hellman in SSH.

FCS_CKM.1(4) Cryptographic Key Generation (for asymmetric keys – IPSec/IKE)

FCS_CKM.1.1(4) **Refinement:** The TSF shall generate **asymmetric** cryptographic keys **used for IKE peer authentication** in accordance with:

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

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

Application Note: This SFR instantiates FSC_CKM.1(2) from VPNGW-EP.

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.

Application Note: “Cryptographic Critical Security Parameters” are defined in FIPS 140-2 as “security-related information (e.g., secret and private cryptographic keys, and authentication data such as passwords and PINs) whose disclosure or modification can compromise the security of a cryptographic module.”

The zeroization indicated above applies to each intermediate storage area for plaintext key/cryptographic critical security parameter (i.e., any storage, such as memory buffers, that is included in the path of such data) upon the transfer of the key/cryptographic critical security parameter to another location.

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* ***GCM***, ***CBC, and CCM*** and cryptographic key sizes *128-bits 256-bits****, and no other key sizes*** that meet the following:

- **FIPS PUB 197, “Advanced Encryption Standard (AES)”**
- **NIST SP 800-38A, NIST SP 800-38C, NIST SP 800-38D**

Application Note: This SFR instantiates the FCS_COP.1(1) requirement from NDPP and VPNGW-EP revised in accordance with NDPP Errata #2.

FCS_COP.1(2) Cryptographic Operation (for cryptographic signature – RSA)

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

RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater

that meets the following:

- **FIPS PUB 186-2 or FIPS PUB 186-3, “Digital Signature Standard”**

Application Note: This SFR instantiates FCS_COP.1(2) RSA/rDSA (which both refer to the RSA Digital Signature Algorithm) requirements from the NDPP and VPNGW-EP.

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.”*

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-bit, 256-bit, 384-bit*** **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.”*

FCS_COP.1(5) Cryptographic Operation (for cryptographic signature – ECDSA)

FCS_COP.1.1(5) **Refinement:** The TSF shall perform **cryptographic signature services** in accordance with a**:**

Elliptic Curve Digital Signature Algorithm (ECDSA) with a key size of 256 bits or greater

that meets the following:

- **FIPS PUB 186-3, “Digital Signature Standard”**
- **The TSF shall implement “NIST curves” P-256, P-384 and no other curves (as defined in FIPS PUB 186-3, “Digital Signature Standard”).**

Application Note: This component is iterated as instructed by the application notes of the NDPP.

Application Note: This SFR instantiates the FCS_COP.1(2) ECDSA requirements from the NDPP and VPNGW-EP.

FCS_RBG_EXT.1(1) Extended: Cryptographic Operation (Random Bit Generation – SSH/TLS)

FCS_RBG_EXT.1.1(1) The TSF shall perform all random bit generation (RBG) services in accordance with FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES seeded by an entropy source that accumulated entropy from a TSF-hardware-based noise source.

FCS_RBG_EXT.1.2 (1) 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.

FCS_RBG_EXT.1(2) Extended: Cryptographic Operation (Random Bit Generation - IPSec)

FCS_RBG_EXT.1.1(2) The TSF shall perform all random bit generation (RBG) services in accordance with NIST Special Publication 800-90 using CTR_DRBG (AES) seeded by an entropy source that accumulated entropy from a TSF-hardware-based noise source.

FCS_RBG_EXT.1.2(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.

Application Note: This SFR instantiates FCS_RBG_EXT.1 from NDPP and VPNGW-EP.

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.

FCS_TLS_EXT.1 Explicit: TLS

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols 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
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

FCS_IPSEC_EXT.1 Explicit: IPSEC

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-CBC128, 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: IKEv1 as defined in RFCs 2407, 2408, 2409, RFC 4109, no other RFCs for extended sequence numbers and RFC 4868 for hash functions; IKEv2 as defined in RFCs 5996 (with mandatory support for NAT traversal as specified in section 2.23) and RFC 4868 for hash functions.

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the IKEv1, IKEv2 protocol uses the cryptographic algorithms AES-CBC-128, AES-CBC-256 as specified in RFC 6379 and no other algorithm.

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

FCS_IPSEC_EXT.1.8 The TSF shall ensure that IKEv2 SA lifetimes can be configured by an Administrator based on number of ~~packets~~ **kb** 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, IKEv1 SA lifetimes can be configured by an Administrator based on number of ~~packets~~ **kb** 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.

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**(2)**, and having a length of at least *384* 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 **of** a specific IPsec SA is less than 1 in 2^ *192 bits*.

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 20 (384-bit Random ECP), no other DH groups.

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 IKEv1 Phase 1, IKEv2 IKE_SA connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the IKEv1 Phase 2, IKEv2 CHILD_SA connection.

Application Note: The FCS_IPSEC_EXT.1 requirement from the VPNGW-EP has been instantiated in place of the NDPP requirement per VPNGW-EP guidance.

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, 4254 and 5656.

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 *32,768* 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 method used for the SSH protocol.

5.3.3 User Data Protection (FDP)

FDP_RIP.2 Full Residual Information Protection

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the *deallocation of the resource from* all objects.

5.3.4 Identification and Authentication (FIA)

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: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “_”, “<”, “>”, “{“, “}”, “[“, “]”, “:”, “.”, “|”, “+”. “~”, “,”, “ ` ”;

2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater;

5.3.5 User Identification and Authentication (FIA_UIA)

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 actions on behalf of that administrative user.

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, *Radius username/password authentication and Public Key authentication*] to perform administrative user authentication.

FIA_UAU.7 Protected Authentication Feedback

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

Application Note: “Obscured feedback” implies the TSF does not produce a visible display of any authentication data entered by a user (such as the echoing of a password), although an obscured indication of progress may be provided (such as an asterisk for each character). It also implies that the TSF does not return any information during the authentication process to the user that may provide any indication of the authentication data.

FIA_AFL.1 Authentication Failure Handling

FIA_AFL.1.1 **Refinement:** The TSF shall detect when **an Administrator configurable positive integer ~~within [assignment: range of acceptable values]~~ 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 Administrator defined time period has elapsed.

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 *between 6 and 64 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 *conversion from ASCII to binary*.

FIA_PSK_EXT.1.4 The TSF shall be able to generate using the random bit generator specified in FCS_RBG_EXT.1 bit-based pre-shared keys.

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 TLS, SSH 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.

5.3.6 Security Management (FMT)

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.

FMT_SMF.1 Specification of Management Functions

FMT_SMF.1.1 **Refinement:** *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;*
- *Ability to configure the cryptographic functionality;*
- *Configure firewall rules;*
- *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~~ **Security Target** to the Administrator;
- *Ability to configure all security management functions identified in other sections of this ~~EP~~ **Security Target**.*

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.

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~~ **Security Target** to an authenticated Administrator.

5.3.7 Protection of the TSF (FPT)

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.

Application Note: The intent of the requirement is that an administrator is unable to read or view the identified keys (stored or ephemeral) through "normal" interfaces. While it is understood that the administrator could directly read memory to view these keys, do so is not a trivial task and may require substantial work on the part of an administrator. Since the administrator is considered a trusted agent, it is assumed they would not endeavour in such an activity.

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.

Application Note: The intent of the requirement is that raw password authentication data are not stored in the clear, and that no user or administrator is able to read the plaintext password through "normal" interfaces. An all-powerful administrator of course could directly read memory to capture a password but is trusted not to do so.

FPT_STM.1 Reliable Time Stamps

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

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.

FPT_TST_EXT.1 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).

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.

5.3.8 TOE Access (FTA)

FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions:

- terminate the session

after a Security Administrator-specified time period of inactivity.

FTA_SSL.3 (1) TSF-initiated Termination (NDPP)

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

FTA_SSL.3(2) TSF-initiated Termination (VPNPP)

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

FTA_SSL.4 User-initiated Termination

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

FTA_TAB.1 Default TOE Access Banners

FTA_TAB.1.1 **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.

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, *blacklist*.

Application Note: Location is defined as the client's IP address. Blacklist refers to a list of denied clients identified by MAC address.

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.

Application Note: The private IP address is one that is internal to the trusted network for which the TOE is the headend.

5.3.9 Trusted Path/Channels (FTP)

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 **authorized IT entities supporting the following capabilities: audit server, authentication server** 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 *Syslog messages and RADIUS authentication*.

Application Note: Authorized IT entities also include VPN Clients and VPN Gateways.

FTP_TRP.1 Trusted Path

FTP_TRP.1.1 **Refinement:** The TSF shall **use SSH, TLS/HTTPS to** 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 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*.

5.3.10 Firewall (FFW)

FFW_RUL_EXT.1 Stateful Traffic Filtering

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

FFW_RUL_EXT.1.2 The TSF shall 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

and distinct interface.

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:

c) 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, *no other attributes*.

d) Remove existing traffic flows from the set of established traffic flows based on the following: session inactivity timeout and/or 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. *SIP*

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,
- *SIP data session in accordance with the SIP protocol specified in RFC 3261*.

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. *The TSF shall reject and be capable of logging network packets where the source IP address is the same as the destination IP address;*

14. *The TSF shall reject and be capable of logging network packets where the protocol field is zero in an IP frame;*

15. *The TSF shall reject and be capable of logging network packets where the length of the frame is not correct. This check is performed for IP, TCP, UDP, GRE, ICMP, PPP and IGMP;*

16. *The TSF shall reject and be capable of logging network packets where the source IP address is all zeros (for non DHCP traffic); and*

17. *The TSF shall reject and be capable of logging network packets where an FTP bounce attack is detected - where dport is equal to FRAME_TCP_PORT_FTP and sport is equal to FRAME_TCP_PORT_FTPD)*.

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.

Application Note: The interfaces (FFW_RUL_EXT.1-5) may be viewed through the CLI command “show interface” as documented in [CLI], or through the WebUI as documented in [USER] Chapter 8.

5.3.11 Packet Filtering (FPF)

FPF_RUL_EXT.1 Packet Filtering

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

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

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

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

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

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

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

and distinct interface.

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

26 The TOE security assurance requirements, summarized in Table 19, are drawn from the NDPP commensurate with EAL1. In accordance with the NDPP, these are supplemented with additional assurance activities as identified at Annex A: NDPP Assurance Activities and sections 4.2 and 4.3 of the TFFW-EP and section 5 of the VPNGW-EP.

Table 19: Assurance Requirements

Assurance Class	Components	Description
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Tests	ATE_IND.1	Independent Testing - conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM Coverage

6 TOE Summary Specification

6.1 Security Functions

6.1.1 Protected Communications

Related SFRs:	FCS_CKM.1(1), FCS_CKM.1(2), FCS_CKM.1(3), FCS_CKM.1(4), FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5), FCS_RBG_EXT.1(1), FCS_RBG_EXT.1(2), FPT_SKP_EXT.1, FTP_ITC.1, FTP_TRP.1, FCS_IPSEC_EXT.1, FCS_SSH_EXT.1, FCS_TLS_EXT.1, FCS_HTTPS_EXT.1, , FIA_PSK_EXT.1, FIA_X509_EXT.1

27 The TOE protects the following communication flows:

a) **WebUI.** Remote administration via the WebUI is protected using TLS/HTTPS.

- TLS/HTTPS is enabled by default.

b) **CLI.** Remote administration via the Command Line Interface (CLI) is protected using SSHv2.

- SSHv2 is enabled by default

c) **Syslog.** Syslog messages are protected using IPSec.

- To set up **site-**to-site IPsec refer to [USER[page 288 “Working with Site-to-Site VPNs”. Configure the IP address of the syslog server as the destination network.

d) **RADIUS.** RADIUS authentication messages are protected using IPSec.

- Same configuration as syslog – set up site-to-site IPsec, and configure the IP address of the RADIUS server as the destination network**.**

e) **VPN.** The TOE can be used as an IPSec VPN gateway. Refer to section 6.1.8.

28 **Note:** The TOE must be operated in a FIPS 140-2 approved mode of operation to ensure that only approved cryptographic operations and algorithms are supported. To enable FIPS mode, use the command “fips enable” from CLI config mode, as documented in the FIPS 140-2 Security Policy. Operation in non-FIPS mode is not part of this evaluation.

29 **Note:** RBG services are not configurable.

30 **Note:** By default, the TOE enables the FTP service for the purpose of providing software images to wireless access points. This service should be disabled when operating in an approved mode of operation. To disable the FTP service, use the CLI command “firewall disable-ftp-server”.

6.1.1.1 TLS\HTTPS

Related SFRs:	FCS_CKM.1(1), FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1(1), FPT_SKP_EXT.1, FTP_TRP.1, FCS_TLS_EXT.1, FCS_HTTPS_EXT.1, FIA_X509_EXT.1

31 The TOE implements a web server that provides the WebUI, The web server is configured by default to use HTTPS. The TOE’s implementation of HTTPS uses TLS 1.2 (RFC 5246) without extensions, supporting the ciphersuites identified in FCS_TLS_EXT.1. The available ciphersuites are not configurable. If the web server

has been configured to use an RSA certificate, the TOE will use RSA-based TLS ciphersuites. If the web server has been configured to use an ECDSA certificate, the TOE will use ECDSA-based TLS ciphersuites.

32 The TOE may be configured to support username/password authentication, client certificate authentication or both.

33 Refer to [USER] Chapter 35 – Management Access. “Configuring Certificate Authentication for WebUI Access” for more information.

34 Certificate stores are described in Section 6.2.4..

6.1.1.2 IPSec

Related SFRs:	FCS_CKM.1(2), FCS_CKM.1(4), FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(3), FCS_COP.1(4), FCS_COP.1(5), FCS_RBG_EXT.1(2), FPT_SKP_EXT.1, FTP_ITC.1, FCS_IPSEC_EXT.1, FIA_PSK_EXT.1, FIA_X509_EXT.1, FIA_X509_EXT.1

35 IPSec is documented in [USER] Chapter 18 “Virtual Private Networks”

36 IPSec can be configured to secure communication with a Syslog server or Radius server. The TOE may also be used as a VPN gateway as described in section 6.1.8 for use with VPN clients or other VPN gateways (i.e. site-to-site VPNs).

37 The TOE’s IPSec implementation has the following characteristics:

a) The algorithms specified at FCS_IPSEC_EXT.1.4 are supported. In addition, AES-CBC-192 and 3DES are also supported by ArubaOS - these algorithms have not been evaluated during the Common Criteria evaluation and must not be used.

b) IKEv1 and IKEv2 are supported.

c) Only tunnel mode is supported. IPsec transport mode is not supported.

d) The “confidentiality only" ESP mode is disabled in the TOE. This behaviour has been hard-coded by excluding the related configuration option from the administrative interfaces (WebUI and CLI).

e) Aggressive mode is not used for IKEv1 Phase 1 exchanges - only main mode is available.

- Aggressive mode must be disabled in order to ensure it is not used. This is documented in [CLI] and is performed using the command “crypto-local isakmp disable-aggressive-mode”.

f) Lifetimes for IKEv1 SAs (both Phase 1 and Phase 2) are established during configuration of the IKE policies by specifying the number of seconds or the number of kb for the SA lifetime.

- Setting the lifetime in number of seconds is documented in [USER} Chapter 18. Volume (traffic) based lifetimes are configured using “crypto dynamic-map set security-association lifetime kilobytes” as documented in [CLI].

g) The TOE supports the DH groups listed at FCS_IPSEC_EXT.1.11. One DH group is configured per IKE policy. IKE policies are incorporated into IPSec maps. IPsec maps are given a priority for peer negotiation. Negotiation requests for security associations will try to match the highest-priority map first. If that map does not match, the negotiation request will continue down the list to the next-highest priority map until a match is made.

h) All IKE protocols implement DH Groups 14 (2048-bit MODP), 19 (256-bit Random ECP), and 20 (384-bit Random ECP.
Note: DH group 2 (1024-bit MODP) is also supported by the TOE – it was not evaluated during the Common Criteria evaluation and must not be used.

   i) For DH Group 14, ‘x’ is generated by an approved DRBG. The IKE process requests an appropriate number of bytes from the DRBG, converts to a “vLong”, and uses the resulting integer value as ‘x’.

   ii) For ECDH (Groups 19/20), ‘x’ is the private key of an ECDSA certificate.

   iii) All nonces and random numbers are generated using a validated RNG/DRBG implementation. For IKEv1 and IKEv2, the TOE uses CTR_DRBG from OpenSSL, FIPS certificate #528/433.

   iv) The length of ‘x’ for DH groups 19/20 is as-specified in the relevant standards. For group 19, it is 256 bits, and for group 20 it is 384 bits. “x” is derived from an ECDSA certificate, which may have been generated within the TOE or may have been externally generated and loaded through a certificate loading function. If the keypair was internally generated, CTR_DRBG from OpenSSL was used as the random number source.

i) IKE peer authentication is performed with either an IKE pre-shared key or digital certificates. IKE policies may be configured to use RSA (rDSA) or ECDSA authentication when using digital certificates. FCS_COP.1(2) requires RSA key sizes of 2048 bits or greater. The TOE supports an RSA key size of 1024 bits in addition to 2048 bits. The administrator must not load an RSA X.509 certificate with a key size smaller than 2048 bits when operating in the Common Criteria evaluated configuration.

j) Pre-shared keys are manually entered during IKE policy configuration. The pre-shared key is used in combination with an agreed DH secret key and an exchanged nonce to generate session keys (SKEYs) which are used to authenticate the two peers to each other as well as to encrypt subsequent IKE exchanges.

k) Pre-shared keys conform to the character and length requirements at FIA_PSK_EXT.1.2.

l) The ASCII representation of pre-shared keys is directly converted to binary.

m) Only HMAC-SHA-1/256/384 are supported, with key and digest sizes of 160, 256, and 384 bits respectively. The TOE prevents configuration of MD5 while operating in FIPS mode.

n) Random number generator services for IPsec are provided automatically by the TOE and do not require administrator configuration.

o) The TOE implements an SPD to determine what traffic gets protected with IPsec, what gets bypassed, and what gets dropped. The SPD is achieved via the routing table and firewall policies. For client-to-gateway VPN links, the firewall policies are in control of how traffic is forwarded. For site-to-site VPN, the routing table is in control. In each case, additional routing or firewall rules may be applied. The TOE administrator implicitly configures the IPsec SPD via the routing table and firewall policies and includes a final rule that causes the network packet to be discarded if no other rules are matched. Packet processing is described in section 6.1.7.

p) The TOE supports AES-128 and AES-256 in GCM or CBC modes for symmetric keys, which have 128 bits and 256 bits of strength, respectively.

The TOE does not perform any checks on strength when negotiating IKEv1 Phase 2 and/or IKEv2 CHILD_SA suites– it is incumbent upon the administrator to configure the appropriate transforms to ensure the desired level of strength. The TOE does not establish an IPsec connection if the administrator configures phase 2 to be greater than phase 1 - the TOE protects this negotiation by default.

38 Certificate stores are described in Section 6.2.4.

6.1.1.3 SSH

Related SFRs:	FCS_CKM.1(3), FCS_CKM_EXT.4, FCS_COP.1(1), FCS_COP.1(3), FCS_COP.1(4), FCS_RBG_EXT.1(1), FPT_SKP_EXT.1, FTP_TRP.1, FCS_SSH_EXT.1, FIA_X509_EXT.1

39 The CLI can be accessed from an SSHv2 enabled client. The TOE's SSH implementation has the following characteristics:

a) SSHv2 is supported

b) Public key and password authentication is supported

- [USER] Chapter 35, "Enabling Public Key Authentication for SSH Access" provides more information.

c) The following algorithms are implemented: SSH_RSA for public keys, AES-CBC-128 and AES-CBC-256 for encryption, HMAC-SHA1 and HMAC-SHA1-96 for integrity. **Note:** The encryption and integrity algorithms used are not configurable by the administrator.

d) Packets greater than 32,768 bytes in an SSH transport connection are dropped.

e) All key exchanges for SSH are performed using DH group 14. This behavior is hard-coded into the TOE.

f) No optional protocol characteristics are implemented.

40 Certificate stores are described in Section 6.2.4.

6.1.2 Verifiable Updates

Related SFRs:	FPT_TUD_EXT.1, FCS_COP.1(2), FCS_COP.1(3)

41 Administrators can update the TOE executable code using image files manually downloaded from the Aruba support portal. The administrator may perform an update from either the WebUI or CLI.

- Upgrade instructions are documented in the release notes for each software release, which will be posted in the same directory as the image file on the support portal.

42 A SHA-256 hash of each update image is digitally signed using Aruba's code signing certificate (RSA 2048 bit). When an update is initiated, the TOE verifies the digital signature with a stored certificate (stored in virtual boot ROM of the VMC).

43 Upon successful verification, the TOE boots using the new image. Should verification fail, the TOE will enter into an error state. The TOE's error state will allow direct console access only, where an administrator can change to a new file partition or TFTP a new image and re-boot.

6.1.3 System Monitoring

Related SFRs:	FAU_GEN.1, FAU_GEN.2, FAU_STG_EXT.1, FPT_STM.1

44 The TOE maintains an audit log of administrative and security relevant events. Logs can optionally be delivered to a Syslog server. The administrator can configure the TOE to protect Syslog messages using IPSec as described in section 6.1.1.2. Further detail regarding Syslog and audit messages is provided in the guidance document: ArubaOS 6.4.2.0-1.3 FIPS Syslog Messages, Ref 0510838-01.

- [USER] Chapter 35, Management Access->Configuring Logging provides more details on configuring to use Syslog. Select all Categories and all Subcategories. Set the logging level to “Warning” for all Categories and Subcategories to generate all of the security event logs as defined in FAU_GEN.1 Table 15.
- If Syslog has been enabled, all audit logs are simultaneously written to both the local audit log and the syslog server.

45 **Note:** The command “show audit-trail” as documented in [CLI] is used to show a log of all administrative actions. By default, only commands which change system behavior are logged. By setting the configuration parameter “audit-trail all”, all commands will be logged including commands which do not alter system behavior.

46 The TOE uses an internal system clock to provide reliable timestamps for audit logs. The system clock can be set manually or by configuring the TOE to use a Network Time Protocol (NTP) server to synchronize its system clock with a central time source. If connectivity to the NTP server is lost, the TOE continues to maintain time using the internal system clock and re-synchronizes with the NTP server once connectivity is re-established.

- [USER] Chapter 35, Management Access->Setting the System Clock provides instructions on setting the system clock.

47 The TOE administrator may configure the firewall to log events by including the ‘log’ keyword when defining a firewall rule.

- [USER] Chapter 19, Roles and Policies->Configuring Firewall Policies provides more information on firewall logging.

48 In the event that a TOE network interface is overwhelmed by traffic the TOE will drop packets. An administrator can examine interface counters (using the ‘show interface’ command) to determine if the TOE has dropped packets due to being overwhelmed by traffic. The TOE does not count or log the overwhelmed packets that are dropped. The packets are dropped inside the Network processor hardware and don’t reach the datapath software. So the “show datapath frame” doesn’t show the dropped overwhelmed packets.

49 The TOE’s local audit log consists of three files (for each audit category) that are 31,768 bytes each. The log files are filled consecutively. Once the last file is full, the TOE will begin overwriting the first log file. The log files may only be access by an Authorized Administrator – described in the following section.

6.1.4 Secure Administration

Related SFRs:	FIA_UIA_EXT.1, FIA_UIA_EXT.2, FIA_PMG_EXT.1, FIA_UAU.7, FIA_AFL.1, FMT_MTD.1, FMT_SMF.1, FMT_SMR.2, FMT_MOF.1, FPT_APW_EXT.1, FTA_SSL_EXT.1, FTA_SSL.3, FTA_SSL.4, FTA_TAB.1, FPT_STM.1

50 Initial configuration of the TOE is performed using a question-and-answer dialog presented through the console port after the TOE is powered on for the first time, or

when the configuration of the TOE has been erased using the “write erase” command. While in this default state, no TOE services are available and the TOE does not forward traffic through network interfaces. During the initial configuration dialog, an administrative username and password is established. Once initial configuration has been completed, the TOE reboots into a secure state.

51 The TOE provides two interfaces for administration: WebUI and CLI. The WebUI is accessed via TLS/HTTPS. The CLI is accessed via SSH or direct console. For both TLS/HTTPS and SSH the TOE can be configured to use username/password only, public key authentication only or both username/password and public key authentication. Direct console to the CLI only supports username/password.

- [USER] Chapter 35 “Management Access” has documentation for setting these options.

52 The TOE can be configured to use a Radius server for username/password authentication. The same user repository (either local or Radius) is used from both WebUI and CLI access. Passwords stored locally are encrypted using the TOE’s KEK and cannot be viewed via any normal interface. Password complexity rules are enforced by the TOE (see FIA_PMG_EXT.1), and passwords are obscured during entry.

- [USER] Chapter 35 – “Enabling RADIUS Server Authentication” and “Implementing a Specific Password Management Policy” provide more instruction on how to configure passwords.
- [USER] Ch. 35 - Implementing a Specific Management Password Policy describes setting minimum password length.

53 The TOE administrator specifies the number of successive failed authentication attempts allowed for remote administrators. When the number of unsuccessful authentication attempts has been met, the TOE prevents the offending remote administrator from successfully authenticating until a defined time period has elapsed.

54 A successful logon takes place when a recognized username/password combination is provided and/or a recognized X.509 client certificate is presented by the administrator’s web browser or SSH client.

55 No administrative functions are accessible prior to administrator log-in. Before establishing an administrative user session the TOE displays an administrator specified advisory notice and consent warning message regarding use of the TOE.

- Banner configuration is documented in the [CLI] under “banner motd”

56 The TOE associates users with their assigned role upon successful authentication. The “Authorized Administrator” role defined by the NDPP equates to the “root” role implemented by the TOE.

57 For both the WebUI and CLI, administrative sessions will terminate according to an administrator defined period of inactivity. The system clock as described in section 6.1.3 is used to time the period of inactivity. Administrators can terminate their own session by logging out.

- [USER] Chapter 35, “Setting an Administrator Session Timeout” provides instructions on setting session timeouts.

The system clock time is also used for timestamps in audit log records. [USER} Chapter 35 - Setting the System Clock describes how the system clock can be changed.

6.1.5 Residual Information Clearing

Related SFRs: FDP_RIP.2

58 The TOE ensures that network packets sent from the TOE do not include data "left over" from the processing of previous network information.

59 The memory buffers used in packet processing are sanitized subsequent to each packet being processed. Buffers are made logically unavailable by overwriting the buffer headers with zeros.

6.1.6 Self Test

Related SFRs: FPT_TST_EXT.1, FPT_FLS.1

60 The TOE performs both power-up and conditional self-tests to verify correct and secure operation. In the event that any self-test fails, the TOE will enter an error state, log the error, and reboot automatically. Failure of self-tests requires reinstalling the OVF template to the virtual machine. Relevant log messages are identified in the following supplements:

a) Aruba 9900 Series Virtual Controller with ArubaOS FIPS Firmware Non-Proprietary Security Policy FIPS 140-2 Level 1 Release Supplement.

61 The following test are performed:

a) ArubaOS OpenSSL Module:
   - i) AES (encrypt/decrypt) KATs
   - ii) Triple-DES (encrypt/decrypt) KATs
   - iii) DRBG KAT
   - iv) RSA (sign/verify) KATs
   - v) ECDSA Pairwise Consistency Test
   - vi) SHA (SHA1, SHA256, SHA384, and SHA512) KATs
   - vii) HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512) KATs

b) ArubaOS Cryptographic Module
   - i) AES (encrypt/decrypt) KATs
   - ii) AES-GCM KAT
   - iii) Triple-DES (encrypt/decrypt) KATs
   - iv) SHA (SHA1, SHA256, SHA384 and SHA512) KAT
   - v) HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512) KAT
   - vi) RSA (sign/verify)
   - vii) ECDSA Pairwise Consistency Test
   - viii) FIPS 186-2 RNG KAT

c) ArubaOS Uboot BootLoader Module
   - i) Firmware Integrity Test: RSA 2048-bit Signature Validation

d) Aruba Hardware Known Answer Tests:
   - i) AES KAT

ii) AES-CCM KAT

iii) AES-GCM KAT

iv) Triple DES KAT

v) HMAC (HMAC-SHA1, HMAC-SHA256, HMAC-SHA384 and HMAC-SHA512) KAT

62 The following Conditional Self-tests are performed by the TOE:

a) **Continuous Random Number Generator Test.** This test is run upon generation of random data by the switch's random number generators to detect failure to a constant value. The module stores the first random number for subsequent comparison, and the module compares the value of the new random number with the random number generated in the previous round and enters an error state if the comparison is successful.

b) **Bypass test.** Ensures that the system has not been placed into a mode of operation where cryptographic operations have been bypassed, without the explicit configuration of the cryptographic officer. To conduct the test, a SHA1 hash of the configuration file is calculated and compared to the last known good hash of the configuration file. If the hashes match, the test is passed. Otherwise, the test fails (indicating possible tampering with the configuration file) and the system is halted.

c) **RSA Pairwise Consistency test.** When the TOE generates a public and private key pair, it carries out pair-wise consistency tests for both encryption and digital signing. The test involves encrypting a randomly-generated message with the public key. If the output is equal to the input message, the test fails. The encrypted message is then decrypted using the private key and if the output is not equal to the original message, the test fails. The same random message is then signed using the private key and then verified with the public key. If the verification fails, the test fails.

d) **ECDSA Pairwise Consistency test.** See above RSA pairwise consistency test description.

e) **Firmware Load Test.** This test is identical to the Uboot BootLoader Module Firmware Integrity Test, except that it is performed at the time a new software image is loaded onto the system. Instead of being performed by the BootLoader, the test is performed by the ArubaOS operating system. If the test fails, the newly loaded software image will not be copied into the image partition, and instead will be deleted. Refer to section 6.1.2.

63 Known-answer tests (KAT) involve operating the cryptographic algorithm on data for which the correct output is already known and comparing the calculated output with the previously generated output (the known answer). If the calculated output does not equal the known answer, the known-answer test shall fail.

64 The above tests are sufficient to demonstrate that the TSF is operating correctly by verifying the integrity of the TSF and the correct operation of cryptographic components.

6.1.7 Firewall

Related SFRs:	FFW_RUL_EXT.1, FPF_RUL_EXT.1.7

All firewall documentation is in [USER] Chapter 19 – Roles and Policies.

65 The TOE performs stateful packet filtering. Filtering rules may be applied to appliance Ethernet interfaces or to user-roles (wireless clients connecting through

APs are placed into user-roles). Stateful packet filter policies are applied to user-roles to allow fine grained control over wireless traffic.

66 The TOE is comprised of a data plane and a control plane. Network packets are processed in the data plane, which is the first component that is initialized. Network interfaces are not brought 'up' until initialization is complete and the data plane operating system is fully initialized. All packet level enforcement is performed within the data plane. In case of system error, packets are dropped by default. The control plane operating system (management interfaces) boots simultaneously.

67 The TOE supports stateful processing of the following protocols:

a) RFC 792 (ICMPv4)

b) RFC 4443 (ICMPv6)

c) RFC 791 (IPv4)

d) RFC 2460 (IPv6)

e) RFC 793 (TCP)

f) RFC 768 (UDP)

g) RFC 959 (FTP)

h) RFC 3261 (SIP)

68 The Aruba Quality Assurance (QA) team performs protocol compliance testing using standards based tools and interoperability testing using a range of external vendor equipment.

69 The TOE allows the definition of a stateful packet filtering policy based on the following attributes that are used to define rules (based on the actions: permit, deny or log) for the associated protocols:

a) ICMPv4
   - i) Type
   - ii) Code

b) ICMPv6
   - i) Type
   - ii) Code

c) IPv4
   - i) Source address
   - ii) Destination Address
   - iii) Transport Layer Protocol

d) IPv6
   - i) Source address
   - ii) Destination Address
   - iii) Transport Layer Protocol

e) TCP
   - i) Source Port
   - ii) Destination Port

f) UDP

i) Source Port

ii) Destination Port

70 The TOE allows the stateful packet filtering rules to be assigned to each distinct network interface. The interfaces may be viewed through the CLI command “show interface” as documented in [CLI], or through the WebUI as documented in [USER] Chapter 8.

71 The TOE is also capable of stateful processing for FTP and SIP. For these protocols, the FTP or SIP control channel is monitored for new DATA connections being established. When a new DATA connection is detected, the corresponding TCP connection is automatically added to the firewall’s datapath forwarding table. Firewall actions will match those applied to the parent TCP connection.

72 Configuration for stateful processing of FTP and SIP is found in [USER] Chapter 19 under “Creating a Network Service Alias”.

73 Stateful session tracking is established and maintained by the data plane operating system thorough inspection of network packets, including handshake transactions.

74 The algorithm applied to incoming packets is as follows:

a) Check for IP fragments and assemble.

b) Parse and identify protocol in the IP packet.

c) Perform length checks and apply default rules.

d) Enforce interface access-lists (ACLs) if configured.

e) Lookup session. If exists, then apply corresponding session / stateful ACLs.

f) Add session if it doesn’t exist. (stateful if the protocol warrants, as listed above.). If stateful also open reverse ports for returning/stateful session. The protocol attributes identified above are used in session determination and will include IP protocol attributes for higher level protocols. TCP processing is further described below. Generate log message, if the ‘log’ keyword is configured in the rule.

g) Derive role for the user and apply role based ACLs. If no role ACLs then apply default ACLs (deny).

h) Perform bandwidth contract enforcement.

i) Perform NATing if required.

75 During TCP session establishment, the session is identified by source IP, destination IP, source port, and destination port. By default, the three-way handshake is not enforced before data is allowed. This behavior can be changed using the configuration “firewall enforce-tcp-handshake”. Once the session has been established in the datapath session table, future packets which match the source IP, destination IP, source port, and destination port are processed by the “fast path” which was previously programmed during session establishment.

76 All configuration commands that begin with “firewall” are global firewall config knobs. Their configuration is found in [USER] Chapter 19 under “Understanding Global Firewall Parameters”

77 By default, TCP sequence numbers are ignored. This behavior can be changed using the configuration “firewall enforce-tcp-sequence”. The reason sequence numbers are ignored by default is that in a standard deployment, the Aruba VMC is used as a wireless LAN controller. Encrypted wireless LAN sessions provide

sequence number checks in the Layer 2 protocol already; adding additional sequence number checks at Layer 3 provides limited additional value while causing a slight performance degradation. For non-encrypted wireless LAN sessions, a sequence number enforcement check is meaningless since all traffic is transmitted clear text over the air – an attacker attempting a man-in-the-middle attack could just as easily insert a “correct” sequence number which would be accepted by the firewall. Where the controller is used as a wired firewall, however, sequence number checks do provide some value and in this case the feature should be enabled.

78 TCP flags are largely ignored by the firewall, with the obvious exception of SYN/ACK during session establishment, and FIN/RST during session teardown.

79 Sessions are removed if the relevant protocol frame is received (e.g. TCP RST) or aged out after a configurable timeout of inactivity (whichever occurs first). Session removal is immediate. Audit log messages are not generated when a session is removed.

80 The TOE enforces the default stateful traffic filtering rules specified at FFW_RUL_EXT.1.8 and any administrator defined rules. These rules, called access-lists (ACLs), are in the sequence specified in the packet processing algorithm above. Only a single access-list may be applied to an Ethernet interface. Multiple access-lists may be applied to a user-role. If multiple access-lists are applied, they are processed in order from top to bottom. The first match found is selected and no further processing takes place. If no match is found, the default action is deny.

6.1.8 VPN Gateway

Related SFRs:	FAU_GEN.1, FCS_CKM.1(2), FCS_CKM.1(4), FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(5), FCS_RBG_EXT.1(2), FCS_IPSEC_EXT.1, FTP_ITC.1, FIA_PSK_EXT.1, FIA_X509_EXT.1, FTA_SSL.3 (2), FTA_TSE.1, FTA_VCM_EXT.1

81 The TOE may be used as a VPN gateway – a device at the edge of a private network that terminates an IPsec tunnel, which provides device authentication, confidentiality, and integrity of information traversing a public or untrusted network. This functionality may be used with VPN clients or with other VPN gateways (i.e. site-to-site VPN).

82 TOE logging is described in section 6.1.3. The TOE logs the events shown in Table 18 which include VPN gateway related events (as mapped to related SFRs).The TOE implements the cryptographic requirements of the VPN gateway as specified in FCS_CKM.1(2), FCS_CKM.1(4), FCS_COP.1(1), FCS_COP.1(2), FCS_COP.1(5), FCS_RBG_EXT.1(2) and FIA_X509_EXT.1. Additional detail regarding the cryptographic components of the TOE is provided in section 6.2.

83 The VPN gateway implements the IPsec protocol which is described in section 6.1.1.2. Further detail is provided in [USER] Chapter 18 “Virtual Private Networks”.

84 When used with VPN clients, the clients initiate the VPN connection. When used with other VPN gateways, the TOE or the other VPN gateway may initiate the VPN connection.

85 The administrator may configure a time-out period after which inactive VPN client sessions will be terminated. The TOE may also be configured to deny establishment of VPN client sessions based on client location (IP address), time, day or blacklisted client MAC address.

86 The TOE assigns a private IP address (internal to the trusted network for which the TOE is the headend) to a VPN client upon successful establishment of a session.

6.2 Cryptography

87 This section incorporates additional detail regarding cryptography required by the NDPP.

88 The TOE uses cryptographic functions provided by FIPS 140-2 validated modules:

- CMVP Certificate #2833

6.2.1 Standards Conformance – Key Generation / Establishment

6.2.1.1 RSA

89 The TOE utilizes RSA for key establishment within HTTPS/TLS and IPSec. The TOE's implementation of RSA conforms to NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography".

90 Sections 5 through 9 of NIST Special Publication 800-56B are applicable to the TOE. The TOE conforms to all shall, shall-not, should and should-not statements. There are no TOE-specific implementation extensions.

6.2.1.2 Diffie-Hellman

91 The TOE utilizes Diffie-Hellman within IPSec and SSH. The TOE's implementation of Diffie-Hellman conforms to NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography".

92 Diffie-Hellman relevant subsections of sections 5 through 8 of NIST Special Publication 800-56A are applicable to the TOE. The TOE conforms to all shall, shall-not, should and should-not statements. There are no TOE-specific implementation extensions.

6.2.1.3 ECDSA

93 The TOE utilizes ECDSA within IPSec. The TOE's implementation of ECDSA conforms to NIST Special Publication 800-56A, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography".

94 Elliptic Curve Cryptography (ECC) relevant subsections of sections 5 through 8 of NIST Special Publication 800-56A are applicable to the TOE. The TOE conforms to all shall, shall-not, should and should-not statements. There are no TOE-specific implementation extensions.

6.2.1.4 IKE authentication key pairs

95 Asymmetric key pairs for IKE authentication are in the form of X.509 certificates, either based on RSA (FIPS 186-4 Appendix B.3) or based on ECDSA p256/p384 (FIPS 186-4 Appendix B.4). Keypairs may be externally generated and loaded by an administrator in the form of a PKCS#12 file. Keypairs may also be generated inside the TOE, during creation of a certificate signing request, in which case the FIPS 186-4 implementation of OpenSSL is used, with RNG input provided by CTR_DRBG. Refer to RSA FIPS algorithm certificates 1528 and 1379 implementation details and which sections of FIPS 186-4 are implemented.

6.2.2 Critical Security Parameters

96 Table 20 below identifies all secret and private keys and Critical Security Parameters (CSPs), the related zeroization procedures and whether any interface is available to view the plaintext key.

97 Note that the plaintext keys identified in Table 20 are not able to be viewed via a ‘normal user interface’, that is, no user interface is provided by design and therefore the keys are protected. Per the NDPP FPT_SKP_EXT.1 application note, it is understood that the administrator could directly read memory to view these keys, [however to] do so is not a trivial task and may require substantial work on the part of an administrator. Since the administrator is considered a trusted agent, it is assumed they would not endeavour in such an activity. Shared secrets entered by a user are only viewable during entry.

Table 20: CSPs

#	Name	Algorithm/Key Size	Generation/Use	Storage	Zeroization
General Keys/CSPs					
1	Key Encryption Key (KEK)	Triple-DES (192 bits)	Hardcoded during manufacturing. Used to protect keys stored in the flash.	Stored in Flash memory (plaintext).	Zeroized by using command ‘write erase all’.
2	DRBG entropy input	SP 800-90a CTR_DRBG (512 bits)	Entropy inputs to DRBG function used to construct the DRBG seed.	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module
3	DRBG seed	SP 800-90a CTR_DRBG (384-bits)	Input to the DRBG that determines the internal state of the DRBG. Generated using DRBG derivation function that includes the entropy input from the entropy source.	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module
4	DRBG Key	SP 800-90a CTR_DRBG (256 bits)	This is the DRBG key used for SP 800-90a CTR_DRBG	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module
5	DRBG V	SP 800-90a CTR_DRBG V (128 bits)	Internal V value used as part of SP 800-90a CTR_DRBG	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module

#	Name	Algorithm/Key Size	Generation/Use	Storage	Zeroization
6	RNG seed	FIPS 186-2 General Purpose RNG seed (512 bits)	Used to seed FIPS approved 186-2 general purpose RNG. Generated from non-approved RNG	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
7	RNG seed key	FIPS 186-2 General Purpose RNG seed key (512 bits)	This is the RNG seed key used for FIPS approved 186-2 general purpose RNG.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
8	Diffie-Hellman private key	Diffie-Hellman (224 bits)	Generated internally by calling FIPS approved RNG during Diffie-Hellman Exchange. Used for establishing DH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
9	Diffie-Hellman public key	Diffie-Hellman (2048 bits) Note: Key size of DH Group 1 (768 bits) and Group 2 (1024 bits) is not allowed in FIPS mode.	Generated internally by calling FIPS approved RNG during Diffie-Hellman Exchange. Used for establishing DH shared secret.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
10	Diffie-Hellman shared secret	Diffie-Hellman (2048 bits)	Established during Diffie-Hellman Exchange. Used for deriving IPSec/IKE cryptographic keys	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
11	EC Diffie-Hellman private key	EC Diffie-Hellman (Curves: P-256 or P-384).	Generated internally by calling FIPS approved RNG during EC Diffie-Hellman Exchange. Used for establishing ECDH shared secret	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module

#	Name	Algorithm/Key Size	Generation/Use	Storage	Zeroization
12	EC Diffie-Hellman public key	EC Diffie-Hellman (Curves: P-256 or P-384).	Generated internally by calling FIPS approved RNG during EC Diffie-Hellman Exchange. Used for establishing ECDH shared secret	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
13	EC Diffie-Hellman shared secret	EC Diffie-Hellman (Curves: P-256 or P-384)	Established during EC Diffie-Hellman Exchange. Used for deriving IPSec/IKE cryptographic keys	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
14	RADIUS server shared secret	8-128 characters shared secret	Entered by CO role. Used for RADIUS server authentication	Stored in SDRAM memory (plaintext).	Zeroized by using command 'write erase all' or by overwriting with a new secret
15	Enable secret	8-64 characters password	Entered by CO role. Used for CO role authentication	Stored in SDRAM memory (plaintext).	Zeroized by using command 'write erase all' or by overwriting with a new secret
16	User Passwords	8-64 characters password	Entered by CO role. Used for User role authentication.	Stored in SDRAM memory (plaintext).	Zeroized by using command 'write erase all' or by overwriting with a new secret
17	RSA Private Key	RSA 2048 bit private key	This key is generated by calling FIPS approved RNG in the module. Used for IKEv1, IKEv2, TLS, OCSP (signing OCSP messages) and EAP-TLS peers authentication.	Stored in Flash memory (plaintext) encrypted with KEK.	Zeroized by using command 'write erase all'

#	Name	Algorithm/Key Size	Generation/Use	Storage	Zeroization
18	RSA public key	RSA 2048 bit public key	This key is generated by calling FIPS approved RNG in the module. Used for IKEv1, IKEv2, TLS, OCSP (verifying OCSP messages) and EAP-TLS peers authentication.	Stored in Flash memory (plaintext) encrypted with KEK.	Zeroized by using command 'write erase all'
19	ECDSA Private Key	ECDSA suite B P-256 or P-384 curves	This key is generated by calling FIPS approved RNG in the module. Used for IKEv1, IKEv2, TLS and EAP-TLS peers authentication.	Stored in Flash memory (plaintext) encrypted with KEK.	Zeroized by using command 'write erase all'
20	ECDSA Public Key	ECDSA suite B P-256 or P-384 curves	This key is generated by calling FIPS approved RNG in the module. Used for IKEv1, IKEv2, TLS and EAP-TLS peers authentication.	Stored in Flash memory (plaintext) encrypted with KEK.	Zeroized by using command 'write erase all'.
IPSec/IKE					
21	IKEv1 Pre-shared key	Shared secret (64 ASCII or 128 HEX characters)	Entered by CO role. Used for IKEv1 peers authentication.	Stored in Flash memory encrypted with KEK.	Zeroized by using command 'write erase all' or by overwriting with a new secret
22	skeyid	Shared Secret (160/256/384 bits)	A shared secret known only to IKE peers. It was established via key derivation function defined in SP800-135 KDF (IKEv1). Used for deriving other keys in IKE protocol implementation.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module.

#	Name	Algorithm/Key Size	Generation/Use	Storage	Zeroization
23	skeyid_d	Shared Secret (160/256/384 bits)	A shared secret known only to IKE peers. It was derived via key derivation function defined in SP800-135 KDF (IKEv1). Used for deriving IKE session authentication key	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
24	SKEYSEED	Shared Secret (160/256/384 bits)	A shared secret known only to IKE peers. It was derived via key derivation function defined in SP800-135 KDF (IKEv2) and it will be used for deriving IKE session authentication key.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
25	IKE session authentication key	HMAC-SHA-1/256/384 (160/256/384 bits)	The IKE session (IKE Phase I) authentication key. This key is derived via key derivation function defined in SP800-135 KDF (IKEv1/IKEv2). Used for IKEv1/IKEv2 payload integrity verification.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
26	IKE session encryption key	Triple-DES (192 bits) /AES/AES-GCM (128/196/256 bits)	The IKE session (IKE Phase I) encrypt key. This key is derived via key derivation function defined in SP800-135 KDF (IKEv1/IKEv2).Used for IKE payload protection.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module

#	Name	Algorithm/Key Size	Generation/Use	Storage	Zeroization
27	IPSec session encryption keys	Triple-DES (192 bits / AES/AES-GCM (128/196/256 bits)	The IPsec (IKE phase II) encryption key. This key is derived via a key derivation function defined in SP800-135 KDF (IKEv1/IKEv2). Used for IPSec traffics protection	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
28	IPSec session authentication keys	HMAC-SHA-1 (160 bits)	The IPsec (IKE Phase II) authentication key. This key is derived via using the KDF defined in SP800-135 KDF (IKEv1/IKEv2). Used for IPSec traffics integrity verification.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
SSHv2					
29	SSHv2 session keys	AES (128/196/256 bits)	This key is derived via a key derivation function defined in SP800-135 KDF (SSHv2). Used for SSHv2 traffics protection.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
30	SSHv2 session authentication key	HMAC-SHA-1 (160-bit)	This key is derived via a key derivation function defined in SP800-135 KDF (SSHv2). Used for SSHv2 traffics integrity verification.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module

#	Name	Algorithm/Key Size	Generation/Use	Storage	Zeroization
TLS					
31	TLS pre-master secret	48 byte secret	This key is transferred into the module, protected by TLS RSA public key.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
32	TLS session encryption key	AES 128/192/256 bits	This key is derived via a key derivation function defined in SP800-135 KDF (TLS). Used for TLS traffics protection.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
33	TLS session authentication key	HMAC-SHA-1/256/384 (160/256/384 bits)	This key is derived via a key derivation function defined in SP800-135 KDF (TLS). Used for TLS traffics integrity verification.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module
SNMPv3					
34	SNMPv3 authentication password	8-64 characters password	Entered by CO role. User for SNMPv3 authentication	Stored in Flash memory (plaintext) encrypted with KEK.	Zeroized by using command 'write erase all' or by overwriting with a new secret
35	SNMPv3 engine ID	8-64 characters password	Entered by CO role. A unique string used to identify the SNMP engine.	Stored in Flash memory (plaintext) encrypted with KEK.	Zeroized by using command 'write erase all' or by overwriting with a new secret
36	SNMPv3 session key	AES-CFB key (128 bits)	This key is derived via a key derivation function defined in SP800-135 KDF (SNMPv3). Used for SNMPv3 traffics protection.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module

#	Name	Algorithm/Key Size	Generation/Use	Storage	Zeroization
802.11i					
37	802.11i Pre-Shared Key (PSK)	Shared secret (8-63 characters)	Entered by CO role. Used for 802.11i client/server authentication	Stored in Flash memory encrypted with KEK.	Zeroized by using command 'write erase all' or by overwriting with a new secret
38	802.11i Pair-Wise Master key (PMK)	Shared secret (256 bits)	The PMK is transferred to the module, protected by IPSec secure tunnel. Used to derive the Pairwise Transient Key (PTK) for 802.11i communications.	Stored in SDRAM (plaintext).	Zeroized by rebooting the module
39	802.11i Pairwise Transient Key (PTK)	Shared secret (512 bits)	This key is used to derive 802.11i session key by using the KDF defined in SP800-108.	Stored in SDRAM memory (plaintext)	Zeroized by rebooting the module
40	802.11i session key	AES-CCM (128 bits)	Derived during 802.11i 4-way handshake by using the KDF defined in SP800-108.	Stored in SDRAM memory (plaintext).	Zeroized by rebooting the module

6.2.3 Roles and Services

6.2.3.1 Crypto Officer Role

The Crypto Officer role has the ability to configure, manage, and monitor all processes and functions within the TOE. Two management interfaces can be used for this purpose:

- SSHv2 CLI

The Crypto Officer can use the CLI to perform non-security-sensitive and security-sensitive monitoring and configuration. The CLI can be accessed remotely by using the SSHv2 secured management session over the Ethernet ports or locally over the serial port. In FIPS mode, the serial port is disabled.

- Web Interface

The Crypto Officer can use the Web Interface as an alternative to the CLI. The Web Interface provides a highly intuitive, graphical interface for a comprehensive set of controller management tools. The Web Interface can be accessed from a TLS-enabled Web browser using HTTPS (HTTP with Secure Socket Layer) on logical port 4343.

See the table below for descriptions of the services available to the Crypto Officer role. Numbers in the "CSP Access" column refers to the Critical Security Parameters table above.

Table 21 - Crypto-Officer Services

Service	Description	Input	Output	CSP Access
SSHv2	Provide authenticated and encrypted remote management sessions while using the CLI	SSHv2 key agreement parameters, SSH inputs, and data	SSHv2 outputs and data	29, 30 (delete)
SNMPv3	Provides ability to query management information	SNMPv3 requests	SNMPv3 responses	34, 35 (read) 36 (delete)
IKEv1/IKEv2-IPSec	Provide authenticated and encrypted remote management sessions to access the CLI functionality	IKEv1/IKEv2 inputs and data; IPSec inputs, commands, and data	IKEv1/IKEv2 outputs, status, and data; IPSec outputs, status, and data	21 (read) 22, 23, 24, 25, 26, 27 and 28 (delete)
Configuring Network Management	Create management Users and set their password and privilege level; configure the SNMP agent	Commands and configuration data	Status of commands and configuration data	34, 35 (read) 36 (delete)
Configuring Module Platform	Define the platform subsystem firmware of the module by entering Bootrom Monitor Mode, File System, fault report, message logging, and other platform related commands	Commands and configuration data	Status of commands and configuration data	None
Configuring Hardware Controllers	Define synchronization features for module	Commands and configuration data	Status of commands and configuration data	None
Configuring Internet Protocol	Set IP functionality	Commands and configuration data	Status of commands and	None

Service	Description	Input	Output	CSP Access
			configuration data	
Configuring Quality of Service (QoS)	Configure QOS values for module	Commands and configuration data	Status of commands and configuration data	None
Configuring VPN	Configure Public Key Infrastructure (PKI); configure the Internet Key Exchange (IKEv1/IKEv2) Security Protocol; configure the IPSec protocol	Commands and configuration data	Status of commands and configuration data	21 (read) 18, 19, 20, 21, 22, 23, 24, 25, 26, 27 and 28 (delete)
Configuring DHCP	Configure DHCP on module	Commands and configuration data	Status of commands and configuration data	None
Configuring Security	Define security features for module, including Access List, Authentication, Authorization and Accounting (AAA), and firewall functionality	Commands and configuration data	Status of commands and configuration data	14, 15, 16 (read/write/delete)
Manage Certificates	Install, rename, and delete X.509 certificates	Commands and configuration data; Certificates and keys	Status of certificates, commands, and configuration	17, 18, 19, 20 (write/delete)
HTTPS over TLS	Secure browser connection over Transport Layer Security acting as a Crypto Officer service (web management interface)	TLS inputs, commands, and data	TLS outputs, status, and data	31, 32 and 33 (delete)

Service	Description	Input	Output	CSP Access
Status Function	Cryptographic officer may use CLI "show" commands or view WebUI via TLS to view the controller configuration, routing tables, and active sessions; view health, temperature, memory status, voltage, and packet statistics; review accounting logs, and view physical interface status	Commands and configuration data	Status of commands and configurations	None
IPSec tunnel establishment for RADIUS protection	Provided authenticated/encrypted channel to RADIUS server	IKEv1/IKEv2 inputs and data; IPSec inputs, commands, and data	IKEv1/IKEv2 outputs, status, and data; IPSec outputs, status, and data	14 and 21 (read/write/delete) 22, 23, 24, 25, 26, 27 and 28 (write/delete)
Self-Test	Perform FIPS start-up tests on demand	None	Error messages logged if a failure occurs	None
Configuring Bypass Operation	Configure bypass operation on the module	Commands and configuration data	Status of commands and configuration data	None
Updating Firmware	Updating firmware on the module	Commands and configuration data	Status of commands and configuration data	None
Configuring Online Certificate Status Protocol (OCSP) Responder	Configuring OCSP responder functionality	OCSP inputs, commands, and data	OCSP outputs, status, and data	29, 30, 31, 32 (read)

Service	Description	Input	Output	CSP Access
Configuring Control Plane Security (CPSec)	Configuring Control Plane Security mode to protect communication with APs using IPSec and issue self signed certificates to APs	Commands and configuration data, IKEv1/IKEv2 inputs and data; IPSec inputs, commands, and data	Status of commands, IKEv1/ IKEv2 outputs, status, and data; IPSec outputs, status, and data and configuration data, self signed certificates	14 and 21 (read/write/delete) 22, 23, 24, 25, 26, 27 and 28 (write/delete)
Zeroization	The cryptographic keys stored in SDRAM memory can be zeroized by rebooting the module. The cryptographic keys (IKEv1 Pre-shared key and 802.11i Pre-Shared Key) stored in the flash can be zeroized by using command 'ap wipe out flash' or by overwriting with a new secret. The other keys/CSPs (KEK, RSA/ECDSA public key/private key and certificate) stored in Flash memory can be zeroized by using command 'write erase all.	Command	Progress information	All CSPs will be destroyed.

6.2.4 Certificate Stores

98 The TOE implements a local certificate store which is used to hold CA certificates (used by the TOE to validate peer/client certificates), server certificates (used by the TOE itself for presentation to peers/clients), and client certificates (used to validate administrators). The certificate store also holds OCSP responder certificates, used to validate OCSP responses when OCSP is configured for the direct trust model.

99 While the TOE is operational, certificates are stored in RAMdisk, in cleartext PEM format along with their associated private key (if applicable). Certificates and private keys are stored in non-volatile flash memory when the system is powered off, encrypted using the Private Key Encryption Key (PKEK) described in the CSP table.

100 The certificate store does not act as a database or as a general-purpose certificate store for any purpose. Usage of specific certificates must be explicitly configured in

the system – i.e. to configure the trusted CA for IPsec, a list of CA certificate names must be explicitly configured under the IPsec section of the configuration.

101 Further detail such as how certificates are loaded into the store are described in the 'Managing Certificates' section of [USER]. Only the authenticated administrator has access to add and remove certificates and there are no commands provided which would export private keys.

102 For outgoing certificates (i.e. certificates that are presented to an IKE or TLS client), the TOE expects a valid server certificate chain to have been previously loaded by the administrator. The TOE makes no attempt to construct a chain out of separately-loaded certificates. If the root CA certificate is present in the certificate file loaded by the administrator, the TOE will transmit this to the client. If the root CA certificate is not present, the TOE will not append it.

103 For incoming certificates (i.e. certificates presented by IKE or TLS clients during authentication), the TOE validates the certificate in accordance with RFC 4158. This is done using the X.509 library provided with OpenSSL, for TLS or IKEv1/IKEv2. The TOE expects that the client has presented a complete certificate chain, including any intermediate CA certificates. It will verify certificate time/date validity, and will verify that the presented certificate chains up to a trusted root CA that has been previously configured by the administrator. Only root CA certificates that have been explicitly loaded by an administrator are trusted – the TOE does not ship with any default root CAs trusted. Optional revocation checking is available using CRL or OCSP – these methods are configured by the administrator. For OCSP, an OCSP responder URL is statically configured for each trusted root by the administrator – the TOE does not read the AIA field from the presented certificate.

104 The TOE performs real-time certificate revocation checks using the Online Certificate Status Protocol (OCSP), or traditional certificate validation using the Certificate Revocation List (CRL) client. These features are described in the 'Understanding OCSP and CRL' section of [USER].

7 Rationale

7.1 Conformance Claim Rationale

105 The following rationale is presented with regard to the PP conformance claims:

a) **TOE type.** As identified in section 2.1, the TOE is a network device, stateful traffic filter firewall and VPN gateway, consistent with the TOE type identified by the NDPP, TFFW-EP and VPNGW-EP.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are identical to those of the NDPP, TFFW-EP and VPNGW-EP (including Appendix D: Requirements for Mobility).

c) **Security objectives.** As shown in section 4, the security objectives are identical to those of the NDPP, TFFW-EP and VPNGW-EP (including Appendix D: Requirements for Mobility).

d) **Security requirements.** As shown in section 5, the security requirements are reproduced from the NDPP, TFFW-EP and VPNGW-EP (including Appendix C: Additional Requirements and Appendix D: Requirements for Mobility). No additional requirements have been specified. In accordance with NDPP section 3.1, footnote 1, FPT_ITT.1 has been excluded as the TOE is not distributed.

7.2 Security Objectives Rationale

106 All security objectives are drawn directly from the NDPP, TFFW-EP and VPNGW-EP (including Appendix D: Requirements for Mobility).

7.3 Security Requirements Rationale

107 All security requirements are drawn directly from the NDPP, TFFW-EP and VPNGW-EP (including Appendix D: Requirements for Mobility).

108 In accordance with NDPP section 3.1, footnote 1, FPT_ITT.1 has been excluded as the TOE is not distributed.

7.4 TOE Summary Specification Rationale

109 Table 22 provides a coverage mapping showing that all SFRs are mapped to the security functions described in the TSS.

Table 22: Map of SFRs to TSS Security Functions

SFR	Protected Communications	Verifiable Updates	System Monitoring	Secure Administration	Residual Information Clearing	Self Test	Firewall	VPN Gateway
FAU_GEN.1			X					X

SFR	Protected Communications	Verifiable Updates	System Monitoring	Secure Administration	Residual Information Clearing	Self Test	Firewall	VPN Gateway
FAU_GEN.2			X					
FAU_STG_EXT.1			X					
FCS_CKM.1(1)	X							
FCS_CKM.1(2)	X	X						X
FCS_CKM.1(3)	X	X						
FCS_CKM.1(4)	X							X
FCS_CKM_EXT.4	X							
FCS_COP.1(1)	X							X
FCS_COP.1(2)	X							X
FCS_COP.1(3)	X							
FCS_COP.1(4)	X							
FCS_COP.1(5)	X							X
FCS_RBG_EXT.1(1)	X							
FCS_RBG_EXT.1(2)	X							X
FCS_HTTPS_EXT.1	X							
FCS_TLS_EXT.1	X							
FCS_IPSEC_EXT.1	X							X
FCS_SSH_EXT.1	X							
FDP_RIP.2					X			
FIA_PMG_EXT.1				X				
FIA_UIA_EXT.1				X				
FIA_UAU_EXT.2				X				

SFR	Protected Communications	Verifiable Updates	System Monitoring	Secure Administration	Residual Information Clearing	Self Test	Firewall	VPN Gateway
FIA_UAU.7				X				
FIA_AFL.1				X				
FIA_PSK_EXT.1	X							X
FIA_X509_EXT.1	X							X
FMT_MTD.1				X				
FMT_SMF.1				X				
FMT_SMR.2				X				
FMT_MOF.1				X				
FPT_SKP_EXT.1	X							
FPT_APW_EXT.1				X				
FPT_STM.1			X	X				
FPT_TUD_EXT.1		X						
FPT_TST_EXT.1						X		
FPT_FLS.1						X		
FTA_SSL_EXT.1				X				
FTA_SSL.3(1)				X				
FTA_SSL.3(2)								X
FTA_SSL.4				X				
FTA_TAB.1				X				
FTA_TSE.1								X
FTA_VCM_EXT.1								X
FTP_ITC.1(1)	X							X

SFR	Protected Communications	Verifiable Updates	System Monitoring	Secure Administration	Residual Information Clearing	Self Test	Firewall	VPN Gateway
FTP_TRP.1	X							
FFW_RUL_EXT.1							X	
FPF_RUL_EXT.1							X	

Annex A: NDPP Assurance Activities

110 The NDPP contains assurance activities that are to be performed in meeting the requirements of the NDPP. As these are spread throughout the NDPP document, the table below provides a consolidated reference.

#	NDPP Source	Requirement	Assurance Family
1.	FAU_GEN.1	The evaluator shall check the administrative guide and ensure that it lists all of the auditable events and provides a format for audit records. Each audit record format type must be covered, along with a brief description of each field. The evaluator shall check to make sure that every audit event type mandated by the PP is described and that the description of the fields contains the information required in FAU_GEN1.2, and the additional information specified in Table 1 of the NDPP.	AGD_OPE
2.	FAU_GEN.1	The evaluator shall also make a determination of the administrative actions that are relevant in the context of this PP. The evaluator shall examine the administrative guide and make a determination of which administrative commands, including subcommands, scripts, and configuration files, are related to the configuration (including enabling or disabling) of the mechanisms implemented in the TOE that are necessary to enforce the requirements specified in the PP. The evaluator shall document the methodology or approach taken while determining which actions in the administrative guide are security relevant with respect to this PP. The evaluator may perform this activity as part of the activities associated with ensuring the AGD_OPE guidance satisfies the requirements.	AGD_OPE
3.	FAU_GEN.1	The evaluator shall test the TOE's ability to correctly generate audit records by having the TOE generate audit records for the events listed in table 1 and administrative actions. This should include all instances of an event--for instance, if there are several different I&A mechanisms for a system, the FIA_UIA_EXT.1 events must be generated for each mechanism. The evaluator shall test that audit records are generated for the establishment and termination of a channel for each of the cryptographic protocols contained in the ST. If HTTPS is implemented, the test demonstrating the establishment and termination of a TLS session can be combined with the test for an HTTPS session. For administrative actions, the evaluator shall test that each action determined by the evaluator above to be security relevant in the context of this PP is auditable. When verifying the test results, the evaluator shall ensure the audit records generated during testing match the format specified in the administrative guide, and that the fields in each audit record have the proper entries. Note that the testing here can be accomplished in conjunction with the testing of the security mechanisms directly. For example, testing performed to ensure that the administrative guidance provided is correct verifies that AGD_OPE.1 is satisfied and should address the invocation of the administrative actions that are needed to verify the audit records are generated as expected.	ATE_IND
4.	FAU_STG_EXT.1	For both types of TOEs (those that act as an audit server and those that send data to an external audit server), there is some amount of local storage. The evaluator shall examine the TSS to ensure it describes the amount of audit data that are stored locally; what happens when the local audit data store is full; and	ASE_TSS AGD_OPE

#	NDPP Source	Requirement	Assurance Family
		how these records are protected against unauthorized access. The evaluator shall also examine the operational guidance to determine that it describes the relationship between the local audit data and the audit data that are sent to the audit log server (for TOEs that are not acting as an audit log server). For example, when an audit event is generated, is it simultaneously sent to the external server and the local store, or is the local store used as a buffer and "cleared" periodically by sending the data to the audit server.	
5.	FAU_STG_ EXT.1.1	**TOE acts as audit server** The evaluator shall examine the TSS to ensure it describes the connection supported from non-TOE entities to send the audit data to the TOE, and how the trusted channel is provided. Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel with the TOE, as well as describe any requirements for other IT entities to connect and send audit data to the TOE (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with other IT entites. The evaluator shall perform the following test for this requirement: Test 1: The evaluator shall establish a session between an external IT entity and the TOE according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the IT entity and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the TOE. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the TOE. The evaluator shall perform this test for each protocol selected in the second selection.	ASE_TSS ATE_IND AGD_OPE
6.	FAU_STG_ EXT.1	**TOE is not an audit server** The evaluator shall examine the TSS to ensure it describes the means by which the audit data are transferred to the external audit server, and how the trusted channel is provided. Testing of the trusted channel mechanism will be performed as specified in the associated assurance activities for the particular trusted channel mechanism. The evaluator shall also examine the operational guidance to ensure it describes how to establish the trusted channel to the audit server, as well as describe any requirements on the audit server (particular audit server protocol, version of the protocol required, etc.), as well as configuration of the TOE needed to communicate with the audit server. The evaluator shall perform the following test for this requirement: Test 1: The evaluator shall establish a session between the TOE and the audit server according to the configuration guidance provided. The evaluator shall then examine the traffic that passes between the audit server and the TOE during several activities of the evaluator's choice designed to generate audit data to be transferred to the audit server. The evaluator shall observe that these data are not able to be viewed in the clear during this transfer, and that they are successfully received by the audit server. The evaluator shall record the particular software (name, version) used on the audit server during testing.	ASE_TSS ATE_IND AGD_OPE
7.	FCS_CKM.	The evaluator shall use the key pair generation portions of "The FIPS 186-3	ATE_IND

#	NDPP Source	Requirement	Assurance Family
	1	Digital Signature Algorithm Validation System (DSA2VS)", "The FIPS 186-3 Elliptic Curve Digital Signature Algorithm Validation System (ECDSA2VS)", and either "The RSA Validation System (RSA2VS)" (for FIPS 186-2) or "The 186-3 RSA Validation System (RSA2VS)" (for FIPS 186-3) as a guide in testing the requirement above, depending on the selection performed by the ST author. This will require that the evaluator have a trusted reference implementation of the algorithms that can produce test vectors that are verifiable during the test. The evaluator shall ensure that the TSS contains a description of how the TSF complies with 800-56A and/or 800-56B, depending on the selections made. This description shall indicate the sections in 800-56A and/or 800-56B that are implemented by the TSF, and the evaluator shall ensure that key establishment is among those sections that the TSF claims to implement. Any TOE-specific extensions, processing that is not included in the documents, or alternative implementations allowed by the documents that may impact the security requirements the TOE is to enforce shall be described.	ASE_TSS
8.	FCS_CKM _EXT.4	The evaluator shall check to ensure the TSS describes each of the secret keys (keys used for symmetric encryption), private keys, and CSPs used to generate key; when they are zeroized (for example, immediately after use, on system shutdown, etc.); and the type of zeroization procedure that is performed (overwrite with zeros, overwrite three times with random pattern, etc.). If different types of memory are used to store the materials to be protected, the evaluator shall check to ensure that the TSS describes the zeroization procedure in terms of the memory in which the data are stored (for example, "secret keys stored on flash are zeroized by overwriting once with zeros, while secret keys stored on the internal hard drive are zeroized by overwriting three times with a random pattern that is changed before each write").	ASE_TSS
9.	FCS_COP. 1(1)	The evaluator shall use tests appropriate to the modes selected in the above requirement from "The Advanced Encryption Standard Algorithm Validation Suite (AESAVS)", "The XTS-AES Validation System (XTSVS)", The CMAC Validation System (CMACVS)", "The Counter with Cipher Block Chaining Message Authentication Code (CCM) Validation System (CCMVS)", and "The Galois/Counter Mode (GCM) and GMAC Validation System (GCMVS)" (these documents are available from http://csrc.nist.gov/groups/STM/cavp/index.html) as a guide in testing the requirement above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.	ATE_IND
10.	FCS_COP. 1(2)	The evaluator shall use the signature generation and signature verification portions of "The Digital Signature Algorithm Validation System" (DSA2VS), "The Elliptic Curve Digital Signature Algorithm Validation System" (ECDSA2VS), and "The RSA Validation System" (RSAVS (for 186-2) or RSA2VS (for 186-3)) as a guide in testing the requirement above. The Validation System used shall comply with the conformance standard identified in the ST (i.e., FIPS PUB 186-2 or FIPS PUB 186-3). This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.	ATE_IND
11.	FCS_COP. 1(3)	The evaluator shall use "The Keyed-Hash Message Authentication Code (HMAC) Validation System (HMACVS)" as a guide in testing the requirement	ATE_IND

#	NDPP Source	Requirement	Assurance Family
		above. This will require that the evaluator have a reference implementation of the algorithms known to be good that can produce test vectors that are verifiable during the test.	
12.	FCS_RBG_EXT.1	Documentation shall be produced—and the evaluator shall perform the activities—in accordance with NDPP Annex D, Entropy Documentation and Assessment. The evaluator shall also perform the following tests, depending on the standard to which the RBG conforms. **Implementations Conforming to FIPS 140-2, Annex C** The reference for the tests contained in this section is The Random Number Generator Validation System (RNGVS) [RNGVS]. The evaluator shall conduct the following two tests. Note that the "expected values" are produced by a reference implementation of the algorithm that is known to be correct. Proof of correctness is left to each Scheme. The evaluator shall perform a Variable Seed Test. The evaluator shall provide a set of 128 (Seed, DT) pairs to the TSF RBG function, each 128 bits. The evaluator shall also provide a key (of the length appropriate to the AES algorithm) that is constant for all 128 (Seed, DT) pairs. The DT value is incremented by 1 for each set. The seed values shall have no repeats within the set. The evaluator ensures that the values returned by the TSF match the expected values. The evaluator shall perform a Monte Carlo Test. For this test, they supply an initial Seed and DT value to the TSF RBG function; each of these is 128 bits. The evaluator shall also provide a key (of the length appropriate to the AES algorithm) that is constant throughout the test. The evaluator then invokes the TSF RBG 10,000 times, with the DT value being incremented by 1 on each iteration, and the new seed for the subsequent iteration produced as specified in NIST-Recommended Random Number Generator Based on ANSI X9.31 Appendix A.2.4 Using the 3-Key Triple DES and AES Algorithms, Section 3. The evaluator ensures that the 10,000th value produced matches the expected value.	Entropy Document ATE_IND
13.	FCS_RBG_EXT.1	**Implementations Conforming to NIST Special Publication 800-90** The evaluator shall perform 15 trials for the RBG implementation. If the RBG is configurable, the evaluator shall perform 15 trials for each configuration. The evaluator shall also confirm that the operational guidance contains appropriate instructions for configuring the RBG functionality. If the RBG has prediction resistance enabled, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) generate a second block of random bits (4) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The next two are additional input and entropy input for the first call to generate. The final two are additional input and entropy input for the second call to generate. These values are randomly generated. “generate one block of random bits” means to generate random bits with number of returned bits equal to the Output Block Length (as defined in NIST SP 800-90). If the RBG does not have prediction resistance, each trial consists of (1) instantiate drbg, (2) generate the first block of random bits (3) reseed, (4)	Entropy Document ATE_IND

#	NDPP Source	Requirement	Assurance Family
		generate a second block of random bits (5) uninstantiate. The evaluator verifies that the second block of random bits is the expected value. The evaluator shall generate eight input values for each trial. The first is a count (0 – 14). The next three are entropy input, nonce, and personalization string for the instantiate operation. The fifth value is additional input to the first call to generate. The sixth and seventh are additional input and entropy input to the call to reseed. The final value is additional input to the second generate call. The following paragraphs contain more information on some of the input values to be generated/selected by the evaluator. **Entropy input:** the length of the entropy input value must equal the seed length. **Nonce:** If a nonce is supported (CTR_DRBG with no df does not use a nonce), the nonce bit length is one-half the seed length. **Personalization string:** The length of the personalization string must be <= seed length. If the implementation only supports one personalization string length, then the same length can be used for both values. If more than one string length is support, the evaluator shall use personalization strings of two different lengths. If the implementation does not use a personalization string, no value needs to be supplied. **Additional input:** the additional input bit lengths have the same defaults and restrictions as the personalization string lengths.	
14.	FCS_HTTPS_EXT.1	The evaluator shall check the TSS to ensure that it is clear on how HTTPS uses TLS to establish an administrative session, focusing on any client authentication required by the TLS protocol vs. Security administrator authentication which may be done at a different level of the processing stack. Testing for this activity is done as part of the TLS testing; this may result in additional testing if the TLS tests are done at the TLS protocol level.	ASE_TSS
15.	FCS_TLS_EXT.1	The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that the ciphersuites supported are specified. The evaluator shall check the TSS to ensure that the ciphersuites specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that TLS conforms to the description in the TSS (for instance, the set of ciphersuites advertised by the TOE may have to be restricted to meet the requirements). The evaluator shall also perform the following test: Test 1: The evaluator shall establish a TLS connection using each of the ciphersuites specified by the requirement. This connection may be established as part of the establishment of a higher-level protocol, e.g., as part of a HTTPS session. It is sufficient to observe the successful negotiation of a ciphersuite to satisfy the intent of the test; it is not necessary to examine the characteristics of the encrypted traffic in an attempt to discern the ciphersuite being used (for example, that the cryptographic algorithm is 128-bit AES and not 256-bit AES). Test 2: The evaluator shall setup a man-in-the-middle tool between the TOE and the TLS Peer and shall perform the following modifications to the traffic: • [Conditional: TOE is a server] Modify at least one byte in the server's nonce in the Server Hello handshake message, and verify that the server denies the client's Finished handshake message.	ASE_TSS ATE_IND

#	NDPP Source	Requirement	Assurance Family
		- [Conditional: TOE is a client] Modify the server's selected ciphersuite in the Server Hello handshake message to be a ciphersuite not presented in the Client Hello handshake message. The evaluator shall verify that the client rejects the connection after receiving the Server Hello. - [Conditional: TOE is a client] If a DHE or ECDHE ciphersuite is supported, modify the signature block in the Server's KeyExchange handshake message, and verify that the client rejects the connection after receiving the Server KeyExchange. [Conditional: TOE is a client] Modify a byte in the Server Finished handshake message, and verify that the client sends a fatal alert upon receipt and does not send any application data.	
16.	FCS_IPSEC_EXT.1	Refer to VPNGW-EP.	
17.	FCS_SSH_EXT.1.2	The evaluator shall check to ensure that the TSS contains a description of the public key algorithms that are acceptable for use for authentication, that this list conforms to FCS_SSH_EXT.1.5, and ensure that password-based authentication methods are also allowed. The evaluator shall also perform the following tests: **Test 1:** The evaluator shall, for each public key algorithm supported, show that the TOE supports the use of that public key algorithm to authenticate a user connection. Any configuration activities required to support this test shall be performed according to instructions in the operational guidance. **Test 2:** Using the operational guidance, the evaluator shall configure the TOE to accept password-based authentication, and demonstrate that a user can be successfully authenticated to the TOE over SSH using a password as an authenticator.	ASE_TSS ATE_IND
18.	FCS_SSH_EXT.1.3	The evaluator shall check that the TSS describes how "large packets" in terms of RFC 4253 are detected and handled. The evaluator shall also perform the following test: **Test 1:** The evaluator shall demonstrate that if the TOE receives a packet larger than that specified in this component, that packet is dropped.	ASE_TSS ATE_IND
19.	FCS_SSH_EXT.1.4	The evaluator shall check the description of the implementation of this protocol in the TSS to ensure that optional characteristics are specified, and the encryption algorithms supported are specified as well. The evaluator shall check the TSS to ensure that the encryption algorithms specified are identical to those listed for this component. The evaluator shall also check the operational guidance to ensure that it contains instructions on configuring the TOE so that SSH conforms to the description in the TSS (for instance, the set of algorithms advertised by the TOE may have to be restricted to meet the requirements). The evaluator shall also perform the following test: **Test 1:** The evaluator shall establish a SSH connection using each of the encryption algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of a protocol to satisfy the intent of the test.	ASE_TSS AGD_OPE ATE_IND

#	NDPP Source	Requirement	Assurance Family
20.	FCS_SSH_ EXT.1.5	The assurance activity associated with FCS_SSH_EXT.1.4 verifies this requirement.	N/A
21.	FCS_SSH_ EXT.1.6	The evaluator shall check the TSS to ensure that it lists the supported data integrity algorithms, and that that list corresponds to the list in this component. The evaluator shall also check the operational guidance to ensure that it contains instructions to the administrator on how to ensure that only the allowed data integrity algorithms are used in SSH connections with the TOE (specifically, that the “none” MAC algorithm is not allowed). The evaluator shall also perform the following test: • Test 1: The evaluator shall establish a SSH connection using each of the integrity algorithms specified by the requirement. It is sufficient to observe (on the wire) the successful negotiation of the algorithm to satisfy the intent of the test.	ASE_TSS
22.	FCS_SSH_ EXT.1.7	The evaluator shall ensure that operational guidance contains configuration information that will allow the security administrator to configure the TOE so that all key exchanges for SSH are performed using DH group 14 and any groups specified from the selection in the ST. If this capability is “hard-coded” into the TOE, the evaluator shall check the TSS to ensure that this is stated in the discussion of the SSH protocol. The evaluator shall also perform the following test: **Test 1:** The evaluator shall attempt to perform a diffie-hellman-group1-sha1 key exchange, and observe that the attempt fails. For each allowed key exchange method the evaluator shall then attempt to perform a key exchange using that method, and observe that the attempt succeeds.	AGD_OPE ASE_TSS ATE_IND
23.	FDP_RIP.2	“Resources” in the context of this requirement are network packets being sent through (as opposed to “to”, as is the case when a security administrator connects to the TOE) the TOE. The concern is that once a network packet is sent, the buffer or memory area used by the packet still contains data from that packet, and that if that buffer is re-used, those data might remain and make their way into a new packet. The evaluator shall check to ensure that the TSS describes packet processing to the extent that they can determine that no data will be reused when processing network packets. The evaluator shall ensure that this description at a minimum describes how the previous data are zeroized/overwritten, and at what point in the buffer processing this occurs.	ASE_TSS
24.	FIA_PMG_ EXT.1	The evaluator shall examine the operational guidance to determine that it provides guidance to security administrators on the composition of strong passwords, and that it provides instructions on setting the minimum password length. The evaluator shall also perform the following tests. Note that one or more of these tests can be performed with a single test case. Test 1: The evaluator shall compose passwords that either meet the requirements, or fail to meet the requirements, in some way. For each password, the evaluator shall verify that the TOE supports the password. While the evaluator is not required (nor is it feasible) to test all possible compositions of passwords, the evaluator shall ensure that all characters, rule characteristics, and a minimum length listed in the requirement are supported, and justify the subset of those characters chosen for testing.	AGD_OPE ATE_IND

#	NDPP Source	Requirement	Assurance Family
25.	FIA_PSK_EXT.1	The evaluator shall examine the operational guidance to determine that it provides guidance on the composition of strong text-based pre-shared keys, and (if the selection indicates keys of various lengths can be entered) that it provides information on the merits of shorter or longer pre-shared keys. The guidance must specify the allowable characters for pre-shared keys, and that list must be a super-set of the list contained in FIA_PSK_EXT.1.2. The evaluator shall examine the TSS to ensure that it states that text-based pre-shared keys of 22 characters are supported, and that the TSS states the conditioning that takes place to transform the text-based pre-shared key from the key sequence entered by the user (e.g., ASCII representation) to the bit string used by IPsec, and that this conditioning is consistent with the first selection in the FIA_PSK_EXT.1.3 requirement. If the assignment is used to specify conditioning, the evaluator will confirm that the TSS describes this conditioning. If "bit-based pre-shared keys" is selected, the evaluator shall confirm the operational guidance contains instructions for either entering bit-based pre-shared keys for each protocol identified in the requirement, or generating a bit-based pre-shared key (or both). The evaluator shall also examine the TSS to ensure it describes the process by which the bit-based pre-shared keys are generated (if the TOE supports this functionality), and confirm that this process uses the RBG specified in FCS_RBG_EXT.1. The evaluator shall also perform the following tests: • Test 1: The evaluator shall compose at least 15 pre-shared keys of 22 characters that cover all allowed characters in various combinations that conform to the operational guidance, and demonstrates that a successful protocol negotiation can be performed with each key. • Test 2 [conditional]: If the TOE supports pre-shared keys of multiple lengths, the evaluator shall repeat Test 1 using the minimum length; the maximum length; and an invalid length. The minimum and maximum length tests should be successful, and the invalid length must be rejected by the TOE. • Test 3 [conditional]: If the TOE supports bit-based pre-shared keys but does not generate such keys, the evaluator shall obtain a bit-based pre-shared key of the appropriate length and enter it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key. • Test 4 [conditional]: If the TOE supports bit-based pre-shared keys and does generate such keys, the evaluator shall generate a bit-based pre-shared key of the appropriate length and use it according to the instructions in the operational guidance. The evaluator shall then demonstrate that a successful protocol negotiation can be performed with the key.	AGD_OPE ASE_TSS ATE_IND
26.	FIA_UIA_EXT.1	The evaluator shall examine the TSS to determine that it describes the logon process for each logon method (local, remote (HTTPS, SSH, etc.)) supported for the product. This description shall contain information pertaining to the credentials allowed/used, any protocol transactions that take place, and what constitutes a "successful logon". The evaluator shall examine the operational	ASE_TSS ATE_IND

#	NDPP Source	Requirement	Assurance Family
		guidance to determine that any necessary preparatory steps (e.g., establishing credential material such as preshared keys, tunnels, certificates, etc.) to logging in are described. For each supported the login method, the evaluator shall ensure the operational guidance provides clear instructions for successfully logging on. If configuration is necessary to ensure the services provided before login are limited, the evaluator shall determine that the operational guidance provides sufficient instruction on limiting the allowed services. The evaluator shall perform the following tests for each method by which administrators access the TOE (local and remote), as well as for each type of credential supported by the login method: **Test 1:** The evaluator shall use the operational guidance to configure the appropriate credential supported for the login method. For that credential/login method, the evaluator shall show that providing correct I&A information results in the ability to access the system, while providing incorrect information results in denial of access. **Test 2:** The evaluator shall configure the services allowed (if any) according to the operational guidance, and then determine the services available to an external remote entity. The evaluator shall determine that the list of services available is limited to those specified in the requirement. **Test 3:** For local access, the evaluator shall determine what services are available to a local administrator prior to logging in, and make sure this list is consistent with the requirement.	
27.	FIA_UAU_ EXT.2	Assurance activities for this requirement are covered under those for FIA_UIA_EXT.1. If other authentication mechanisms are specified, the evaluator shall include those methods in the activities for FIA_UIA_EXT.1.	ASE_TSS ATE_IND
28.	FIA_UAU.7	The evaluator shall perform the following test for each method of local login allowed: **Test 1:** The evaluator shall locally authenticate to the TOE. While making this attempt, the evaluator shall verify that at most obscured feedback is provided while entering the authentication information.	ATE_IND
29.	FMT_MTD.1	The evaluator shall review the operational guidance to determine that each of the TSF-data-manipulating functions implemented in response to the requirements of this PP is identified, and that configuration information is provided to ensure that only administrators have access to the functions. The evaluator shall examine the TSS to determine that, for each administrative function identified in the operational guidance; those that are accessible through an interface prior to administrator log-in are identified. For each of these functions, the evaluator shall also confirm that the TSS details how the ability to manipulate the TSF data through these interfaces is disallowed for non-administrative users.	AGD_OPE ASE_TSS
30.	FMT_SMF.1	The security management functions for FMT_SMF.1 are distributed throughout the PP and are included as part of the requirements in FMT_MTD, FPT_TST_EXT, and any cryptographic management functions specified in the reference standards. Compliance to these requirements satisfies compliance with FMT_SMF.1.	N/A

#	NDPP Source	Requirement	Assurance Family
31.	FMT_SMR.2	The evaluator shall review the operational guidance to ensure that it contains instructions for administering the TOE both locally and remotely, including any configuration that needs to be performed on the client for remote administration. In the course of performing the testing activities for the evaluation, the evaluator shall use all supported interfaces, although it is not necessary to repeat each test involving an administrative action with each interface. The evaluator shall ensure, however, that each supported method of administering the TOE that conforms to the requirements of this PP be tested; for instance, if the TOE can be administered through a local hardware interface; SSH; and TLS/HTTPS; then all three methods of administration must be exercised during the evaluation team's test activities.	AGD_OPE ATE_IND
32.	FPT_SKP_EXT.1	The evaluator shall examine the TSS to determine that it details how any pre-shared keys, symmetric keys, and private keys are stored and that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note. If these values are not stored in plaintext, the TSS shall describe how they are protected/obscured.	ASE_TSS
33.	FPT_APW_EXT.1	The evaluator shall examine the TSS to determine that it details all authentication data that are subject to this requirement, and the method used to obscure the plaintext password data when stored. The TSS shall also detail passwords are stored in such a way that they are unable to be viewed through an interface designed specifically for that purpose, as outlined in the application note.	ASE_TSS
34.	FPT_STM.1	The evaluator shall examine the TSS to ensure that it lists each security function that makes use of time. The TSS provides a description of how the time is maintained and considered reliable in the context of each of the time related functions. The evaluator examines the operational guidance to ensure it instructs the administrator how to set the time. If the TOE supports the use of an NTP server, the operational guidance instructs how a communication path is established between the TOE and the NTP server, and any configuration of the NTP client on the TOE to support this communication. **Test 1:** The evaluator uses the operational guide to set the time. The evaluator shall then use an available interface to observe that the time was set correctly. **Test2:** [conditional] If the TOE supports the use of an NTP server; the evaluator shall use the operational guidance to configure the NTP client on the TOE, and set up a communication path with the NTP server. The evaluator will observe that the NTP server has set the time to what is expected. If the TOE supports multiple protocols for establishing a connection with the NTP server, the evaluator shall perform this test using each supported protocol claimed in the operational guidance.	ASE_TSS AGD_OPE ATE_IND
35.	FPT_TUD_EXT.1	Updates to the TOE either have a hash associated with them, or are signed by an authorized source. If digital signatures are used, the definition of an authorized source is contained in the TSS, along with a description of how the certificates used by the update verification mechanism are contained on the device. The evaluator ensures this information is contained in the TSS. The evaluator also ensures that the TSS (or the operational guidance) describes how	ASE_TSS AGD_OPE ATE_IND

#	NDPP Source	Requirement	Assurance Family
		the candidate updates are obtained; the processing associated with verifying the digital signature or calculating the hash of the updates; and the actions that take place for successful (hash or signature was verified) and unsuccessful (hash or signature could not be verified) cases. The evaluator shall perform the following tests: **Test 1:** The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains a legitimate update using procedures described in the operational guidance and verifies that it is successfully installed on the TOE. Then, the evaluator performs a subset of other assurance activity tests to demonstrate that the update functions as expected. After the update, the evaluator performs the version verification activity again to verify the version correctly corresponds to that of the update. **Test 2:** The evaluator performs the version verification activity to determine the current version of the product. The evaluator obtains or produces an illegitimate update, and attempts to install it on the TOE. The evaluator verifies that the TOE rejects the update.	
36.	FPT_TST_ EXT.1	The evaluator shall examine the TSS to ensure that it details the self tests that are run by the TSF on start-up; this description should include an outline of what the tests are actually doing (e.g., rather than saying "memory is tested", a description similar to "memory is tested by writing a value to each memory location and reading it back to ensure it is identical to what was written" shall be used). The evaluator shall ensure that the TSS makes an argument that the tests are sufficient to demonstrate that the TSF is operating correctly. The evaluator shall also ensure that the operational guidance describes the possible errors that may result from such tests, and actions the administrator should take in response; these possible errors shall correspond to those described in the TSS.	ASE_TSS AGD_OPE
37.	FTA_SSL_ EXT.1	The evaluator shall perform the following test: **Test 1:** The evaluator follows the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a local interactive session with the TOE. The evaluator then observes that the session is either locked or terminated after the configured time period. If locking was selected from the component, the evaluator then ensures that re-authentication is needed when trying to unlock the session.	ATE_IND
38.	FTA_SSL.3	The evaluator shall perform the following test: **Test 1:** The evaluator follows the operational guidance to configure several different values for the inactivity time period referenced in the component. For each period configured, the evaluator establishes a remote interactive session with the TOE. The evaluator then observes that the session is terminated after the configured time period.	ATE_IND
39.	FTA_SSL.4	The evaluator shall perform the following test: **Test 1:** The evaluator initiates an interactive local session with the TOE. The evaluator then follows the operational guidance to exit or log off the session and observes that the session has been terminated.	ATE_IND

#	NDPP Source	Requirement	Assurance Family
		Test 2: The evaluator initiates an interactive remote session with the TOE. The evaluator then follows the operational guidance to exit or log off the session and observes that the session has been terminated.	
40.	FTA_TAB.1	The evaluator shall check the TSS to ensure that it details each method of access (local and remote) available to the administrator (e.g., serial port, SSH, HTTPS). The evaluator shall also perform the following test: **Test 1:** The evaluator follows the operational guidance to configure a notice and consent warning message. The evaluator shall then, for each method of access specified in the TSS, establish a session with the TOE. The evaluator shall verify that the notice and consent warning message is displayed in each instance.	ATE_TSS ATE_IND
41.	FTP_ ITC.1	The evaluator shall examine the TSS to determine that, for all communications with authorized IT entities identified in the requirement, each communications mechanism is identified in terms of the allowed protocols for that IT entity. The evaluator shall also confirm that all protocols listed in the TSS are specified and included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the allowed protocols with each authorized IT entity, and that it contains recovery instructions should a connection be unintentionally broken. The evaluator shall also perform the following tests: **Test 1:** The evaluators shall ensure that communications using each protocol with each authorized IT entity is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful. **Test 2:** For each protocol that the TOE can initiate as defined in the requirement, the evaluator shall follow the operational guidance to ensure that in fact the communication channel can be initiated from the TOE. **Test 3:** The evaluator shall ensure, for each communication channel with an authorized IT entity, the channel data are not sent in plaintext. **Test 4:** The evaluators shall, for each protocol associated with each authorized IT entity tested during test 1, the connection is physically interrupted. The evaluator shall ensure that when physical connectivity is restored, communications are appropriately protected. Further assurance activities are associated with the specific protocols.	ASE_TSS AGD_OPE ATE_IND
42.	FTP_TRP.1	The evaluator shall examine the TSS to determine that the methods of remote TOE administration are indicated, along with how those communications are protected. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the remote administrative sessions for each supported method. The evaluator shall also perform the following tests: **Test 1:** The evaluators shall ensure that communications using each specified (in the operational guidance) remote administration method is tested during the course of the evaluation, setting up the connections as described in the	ASE_TSS

#	NDPP Source	Requirement	Assurance Family
		operational guidance and ensuring that communication is successful. **Test 2:** For each method of remote administration supported, the evaluator shall follow the operational guidance to ensure that there is no available interface that can be used by a remote user to establish a remote administrative sessions without invoking the trusted path. **Test 3:** The evaluator shall ensure, for each method of remote administration, the channel data is not sent in plaintext. Further assurance activities are associated with the specific protocols.	
43.	FPT_ITT.1	The evaluator shall examine the TSS to determine that the methods and protocols used to protect distributed TOE components are described. The evaluator shall also confirm that all protocols listed in the TSS in support of TOE administration are consistent with those specified in the requirement, and are included in the requirements in the ST. The evaluator shall confirm that the operational guidance contains instructions for establishing the communication paths for each supported method. The evaluator shall also perform the following tests: **Test 1:** The evaluators shall ensure that communications using each specified (in the operational guidance) communications method is tested during the course of the evaluation, setting up the connections as described in the operational guidance and ensuring that communication is successful. **Test 2:** The evaluator shall ensure, for each method of communication, the channel data is not sent in plaintext. **Test 3:** The evaluator shall ensure, for each method of communication, modification of the channel data is detected by the TOE. Further assurance activities are associated with the specific protocols.	ASE_TSS ATE_IND
44.	ADV_FSP.1	Developer Note: As indicated in the introduction to this section, the functional specification is comprised of the information contained in the AGD_OPE and AGD_PRE documentation, coupled with the information provided in the TSS of the ST. The assurance activities in the functional requirements point to evidence that should exist in the documentation and TSS section; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.	ADV_FSP
45.	ADV_FSP.1	There are no specific assurance activities associated with these SARs. The functional specification documentation is provided to support the evaluation activities described in NDPP Section 4.2, and other activities described for AGD, ATE, and AVA SARs. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed; if the evaluator is unable to perform an activity because the there is insufficient interface information, then an adequate functional specification has not been provided.	ADV_FSP
46.	AGD_OPE.1	The operational guidance shall at a minimum list the processes running (or that could run) on the TOE in its evaluated configuration during its operation that are capable of processing data received on the network interfaces (there are likely	AGD_OPE

#	NDPP Source	Requirement	Assurance Family
		more than one of these, and this is not limited to the process that "listens" on the network interface). It is acceptable to list all processes running (or that could run) on the TOE in its evaluated configuration instead of attempting to determine just those that process the network data. For each process listed, the administrative guidance will contain a short (e.g., one- or two-line) description of the process' function, and the privilege with which the service runs. "Privilege" includes the hardware privilege level (e.g., ring 0, ring 1), any software privileges specifically associated with the process, and the privileges associated with the user role the process runs as or under.	
47.	AGD_OPE.1	The operational guidance shall contain instructions for configuring the cryptographic engine associated with the evaluated configuration of the TOE. It shall provide a warning to the administrator that use of other cryptographic engines was not evaluated nor tested during the CC evaluation of the TOE.	AGD_OPE
48.	AGD_OPE.1	The documentation must describe the process for verifying updates to the TOE, either by checking the hash or by verifying a digital signature. The evaluator shall verify that this process includes the following steps: 1. For hashes, a description of where the hash for a given update can be obtained. For digital signatures, instructions for obtaining the certificate that will be used by the FCS_COP.1(2) mechanism to ensure that a signed update has been received from the certificate owner. This may be supplied with the product initially, or may be obtained by some other means. 2. Instructions for obtaining the update itself. This should include instructions for making the update accessible to the TOE (e.g., placement in a specific directory). 3. Instructions for initiating the update process, as well as discerning whether the process was successful or unsuccessful. This includes generation of the hash/digital signature.	AGD_OPE
49.	AGD_OPE.1	The TOE will likely contain security functionality that does not fall in the scope of evaluation under the NDPP. The operational guidance shall make it clear to an administrator which security functionality is covered by the evaluation activities.	AGD_OPE
50.	AGD_PRE.1	The evaluator shall check to ensure that the guidance provided for the TOE adequately addresses all platforms claimed for the TOE in the ST.	AGD_PRE
51.	ATE_IND.1	The evaluator shall prepare a test plan and report documenting the testing aspects of the system. The test plan covers all of the testing actions contained in the CEM and the body of the NDPP's Assurance Activities. While it is not necessary to have one test case per test listed in an Assurance Activity, the evaluator must document in the test plan that each applicable testing requirement in the ST is covered.	ATE_IND
52.	ATE_IND.1	The test plan identifies the platforms to be tested, and for those platforms not included in the test plan but included in the ST, the test plan provides a justification for not testing the platforms. This justification must address the differences between the tested platforms and the untested platforms, and make an argument that the differences do not affect the testing to be performed. It is not sufficient to merely assert that the differences have no affect; rationale must be provided. If all platforms claimed in the ST are tested, then no rationale is	ATE_IND

#	NDPP Source	Requirement	Assurance Family
		necessary.	
53.	ATE_IND.1	The test plan describes the composition of each platform to be tested, and any setup that is necessary beyond what is contained in the AGD documentation. It should be noted that the evaluator is expected to follow the AGD documentation for installation and setup of each platform either as part of a test or as a standard pre-test condition. This may include special test drivers or tools. For each driver or tool, an argument (not just an assertion) should be provided that the driver or tool will not adversely affect the performance of the functionality by the TOE and its platform. This also includes the configuration of the cryptographic engine to be used. The cryptographic algorithms implemented by this engine are those specified by this PP and used by the cryptographic protocols being evaluated (IPsec, TLS/HTTPS, SSH).	ATE_IND
54.	ATE_IND.1	The test plan identifies high-level test objectives as well as the test procedures to be followed to achieve those objectives. These procedures include expected results. The test report (which could just be an annotated version of the test plan) details the activities that took place when the test procedures were executed, and includes the actual results of the tests. This shall be a cumulative account, so if there was a test run that resulted in a failure; a fix installed; and then a successful rerun of the test, the report would show a “fail” and “pass” result (and the supporting details), and not just the “pass” result.	ATE_IND
55.	AVA_VAN.1	As with ATE_IND, the evaluator shall generate a report to document their findings with respect to this requirement. This report could physically be part of the overall test report mentioned in ATE_IND, or a separate document. The evaluator performs a search of public information to determine the vulnerabilities that have been found in network infrastructure devices and the implemented communication protocols in general, as well as those that pertain to the particular TOE. The evaluator documents the sources consulted and the vulnerabilities found in the report. For each vulnerability found, the evaluator either provides a rationale with respect to its nonapplicability, or the evaluator formulates a test (using the guidelines provided in ATE_IND) to confirm the vulnerability, if suitable. Suitability is determined by assessing the attack vector needed to take advantage of the vulnerability. For example, if the vulnerability can be detected by pressing a key combination on boot-up, a test would be suitable at the assurance level of the NDPP. If exploiting the vulnerability requires expert skills and an electron microscope, for instance, then a test would not be suitable and an appropriate justification would be formulated.	AVA_VAN
56.	ALC_CMC.1	The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. The evaluator shall ensure that this identifier is sufficient for an acquisition entity to use in procuring the TOE (including the appropriate administrative guidance) as specified in the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST. If the vendor maintains a web site advertising the TOE, the evaluator shall examine the information on the web site to ensure that the information in the ST is sufficient to distinguish the product.	ALC_CMC
57.	ALC_CMS.	The “evaluation evidence required by the SARs” in the NDPP is limited to the	ALC_CMS

#	NDPP Source	Requirement	Assurance Family
	2	information in the ST coupled with the guidance provided to administrators and users under the AGD requirements. By ensuring that the TOE is specifically identified and that this identification is consistent in the ST and in the AGD guidance (as done in the assurance activity for ALC_CMC.1), the evaluator implicitly confirms the information required by this component.	
58.	Annex C1.2	The evaluator shall check to ensure that the TSS contains a list (possibly empty except for authentication failures for user-level connections) of the protocol failures that are auditable. The evaluator shall test all identified audit events during protocol testing/audit testing.	ASE_TSS ATE_IND

----- End of Document -----