HPE Aruba Networking Orchestrator and EdgeConnect Release 9.4.2

Security Target

Version 1.16

14 March 2025

Table of Contents

List of Tables

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the HPE Aruba Networking Orchestrator and EdgeConnect Release 9.4.2 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 HPE Aruba Networking EdgeConnect provides SD-WAN, firewall, segmentation, routing, WAN optimization and application visibility and control in one centrally managed platform. Aruba Orchestrator provides management of EdgeConnect appliances giving enterprises the ability to centrally assign policies to secure and control applications across the WAN.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	HPE Aruba Networking Orchestrator and EdgeConnect Release 9.4.2
Security Target	HPE Aruba Networking Orchestrator and EdgeConnect Release 9.4.2 Security Target, v1.16

1.3 Conformance Claims

1.3.1 CC Conformance Claim

3 The TOE and ST are conformant to the following:

a) CC Part 1, Version 3.1, Revision 5

b) CC Part 2, Version 3.1, Revision 5 (extended)

c) CC Part 3 Version 3.1, Revision 5 (conformant)

4 The TOE and the ST are package conformant to the following: **none**

1.3.2 PP Conformance Claim

5 This TOE is Protection Profile conformant to the following:

i) collaborative Protection Profile for Network Devices, Version 2.2e, 23-March-2020 (CPP_ND_V2.2E)

ii) PP-Module for Stateful Traffic Filter Firewalls, Version 1.4 + Errata 20200625, 25-June-2020 (MOD_CPP_FW_V1.4e)

iii) PP-Module for Virtual Private Network (VPN) Gateways, Version 1.3, 16-August-2023 (MOD_VPNGW_V1.3)

6 The conformance is claimed in accordance with PP-Configuration for Network Devices, Stateful Traffic Filter Firewalls, and Virtual Private Network (VPN) Gateways, Version 1.3, 18-August-2023 (CFG_NDcPP-FW-VPNGW_V1.3)

1.3.3 Conformance Claim Rationale

7 This Security Target claims exact conformance to CPP_ND_V2.2E, MOD_CPP_FW_V1.4e, and MOD_VPNGW_V1.3 in accordance with CFG_NDcPP-FW-VPNGW_V1.3. The security problem definition, security objectives and security requirements in this Security Target are all taken from the Protection Profiles and follow the wordings exactly. Only those operations that are allowed in the Protection Profiles are performed on the Security Functional Requirements.

8 The ST claims strict conformance to the Protection Profiles which do not implement any assurance package but explicitly state the applicable security assurance requirements without a reference to the Evaluation Assurance Levels of Common Criteria. Therefore, the ST does not claim package conformance to any assurance package. As per CFG_NDcPP-FW-VPNGW_V1.3, the Security Assurance Components stated in the Base-PP apply to the entire TOE.

1.3.4 Technical Decisions

9 The NIAP Technical Decisions (TD) on CPP_ND_V2.2E, MOD_CPP_FW_V1.4e, and MOD_VPNGW_V1.3, and their applicability to the TOE are given in Table 2. When a TD is not applicable, an exclusion rationale is given.

Table 2: NIAP Technical Decisions

TD #	Name	Rationale if n/a
TD0527	Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	
TD0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	
TD0536	NIT Technical Decision for Update Verification Inconsistency	
TD0537	NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	
TD0545	NIT Technical Decision for Conflicting FW rules cannot be configured (extension of Rfl#201837)	
TD0546	NIT Technical Decision for DTLS – clarification of Application Note 63	FCS_DTLSC_EXT.1 Not claimed
TD0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	
TD0551	NIT Technical Decision for Incomplete Mappings of Oes in FW Module v1.4+Errata	
TD0555	NIT Technical Decision for RFC Reference incorrect in TLSS Test	
TD0556	NIT Technical Decision for RFC 5077 question	

TD #	Name	Rationale if n/a
TD0563	NiT Technical Decision for Clarification of audit date information	
TD0564	NiT Technical Decision for Vulnerability Analysis Search Criteria	
TD0569	NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	FCS_DTLSS_EXT.1 is not claimed
TD0570	NiT Technical Decision for Clarification about FIA_AFL.1	
TD0571	NiT Technical Decision for Guidance on how to handle FIA_AFL.1	
TD0572	NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	
TD0580	NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	
TD0581	NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	
TD0591	NIT Technical Decision for Virtual TOEs and hypervisors	The virtual TOE component is not evaluated as a physical network device
TD0592	NIT Technical Decision for Local Storage of Audit Records	
TD0631	NIT Technical Decision for Clarification of public key authentication for SSH Server	
TD0632	NIT Technical Decision for Consistency with Time Data for vNDs	
TD0635	NIT Technical Decision for TLS Server and Key Agreement Parameters	FCS_TLSS_EXT.1 is not claimed
TD0636	NIT Technical Decision for Clarification of Public Key User Authentication for SSH	FCS_SSHC_EXT.1 Not Claimed
TD0638	NIT Technical Decision for Key Pair Generation for Authentication	
TD0639	NIT Technical Decision for Clarification for NTP MAC Keys	

TD #	Name	Rationale if n/a
TD0670	NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	FCS_TLSC_EXT.1 is not claimed
TD0738	NIT Technical Decision for Link to Allowed-With List	
TD0781	Correction to FIA_PSK_EXT.3 EA for MOD_VPNGW_V1.3_v1.3	FIA_PSK_EXT.3 Not Claimed
TD0790	NIT Technical Decision: Clarification Required for testing IPv6	FCS_DTLSC_EXT.1 and FCS_TLSC_EXT.1 are not claimed
TD0792	NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	
TD0800	Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	
TD0811	Correction to Referenced SFR in FIA_PSK_EXT.3 Test	FIA_PSK_EXT.3 Not Claimed
TD0824	Aligning MOD_VPNGW_V1.3 1.3 with NDcPP 3.0E	Aligning MOD_VPNGW 1.3 with NDcPP 3.0E
TD0827	Aligning MOD_CPP_FW_V1.4eE with CPP_ND_V3.0E	
TD0838	PPK Configurability in FIA_PSK_EXT.1.1	

1.4 Terminology

Table 3: Terminology

Term	Definition
CC	Common Criteria
EAL	Evaluation Assurance Level
ITT	Inter-TSF Trusted Channel
NDcPP	collaborative Protection Profile for Network Devices
PP	Protection Profile
SD-WAN	Software Defined Wide Area Network
ST	Security Target
TOE	Target of Evaluation

Term	Definition
TSF	TOE Security Functionality
WAN	Wide Area Network

2 TOE Description

2.1 Type

10 The TOE is a distributed network device that provides VPN Gateway and Firewall capabilities.

2.2 Usage

2.2.1 Deployment

11 Orchestrator is deployed as an on-premises virtual appliance to monitor and manage one or more EdgeConnects. EdgeConnect devices are deployed on the network edge as gateway devices providing firewall and VPN services. The Orchestrator manages one or more EdgeConnect devices and EdgeConnect devices communicate between each other via VPN tunneling.

2.2.2 Interfaces

12 The TOE interfaces are shown in Figure 1.

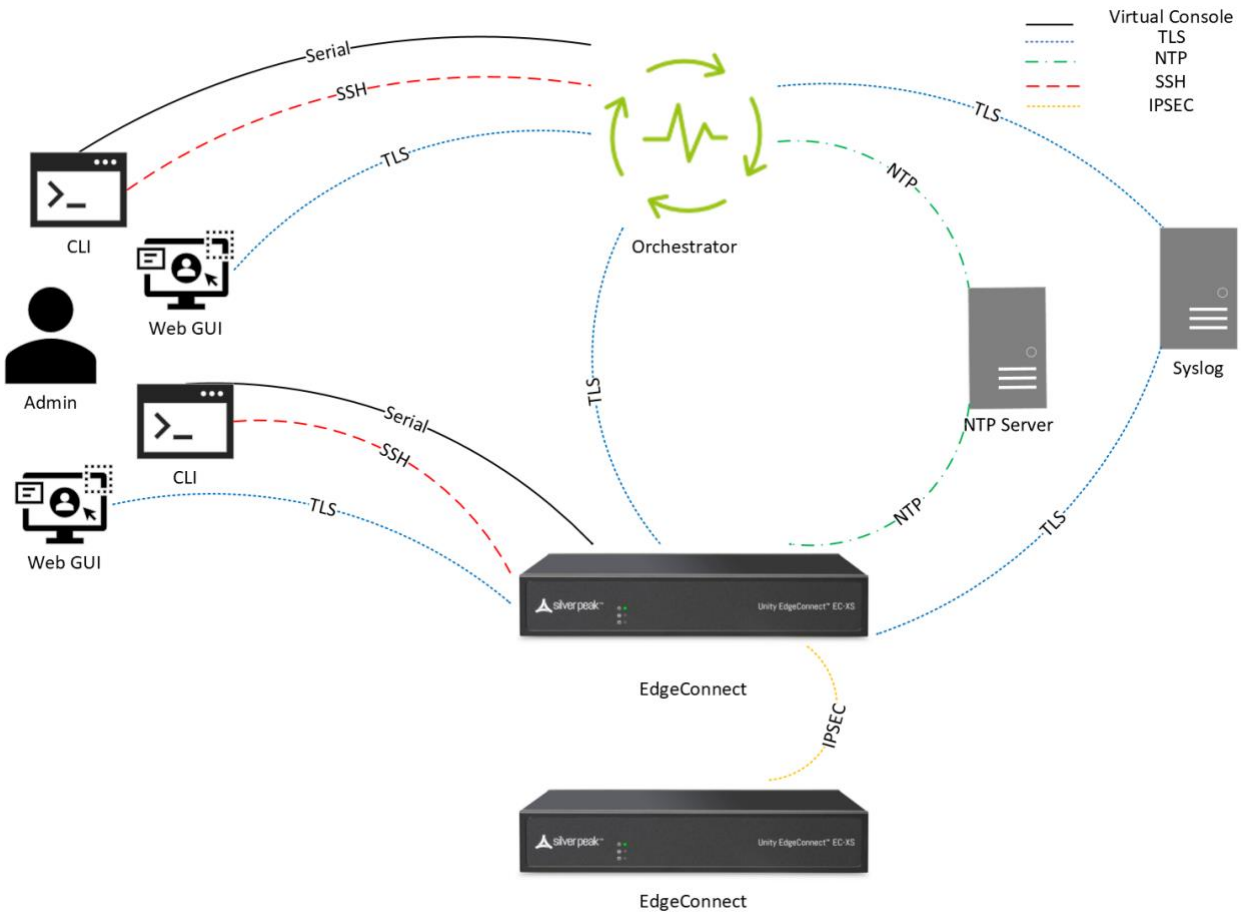

Figure 1: TOE interfaces

13 The TOE interfaces are as follows:

a) **Orchestrator CLI.** Command line management interface via virtual console or remote SSH.

b) **Orchestrator Web GUI.** HTTPS Web management interface via TLS

c) **Orchestrator Syslog.** Transmission of logs to a remote server via TLS.

d) **Orchestrator NTP.** Time updates via NTP.

e) **Orchestrator to EdgeConnect.** Inter TOE management connection via TLS.

f) **EdgeConnect CLI.** Command line management interface via serial console or remote SSH.

g) **EdgeConnect Web GUI.** HTTPS Web management interface via TLS

h) **EdgeConnect Syslog.** Transmission of logs to a remote server via TLS.

i) **EdgeConnect NTP.** Time updates via NTP.

j) **EdgeConnect VPN Tunnel.** Encrypted VPN tunnel between EdgeConnects via IPsec.

2.3 Security Functions / Logical Scope

14 The TOE provides the following security functions:

a) **Protected Communications.** The TOE protects the integrity and confidentiality of communications as noted in section 2.2.2 above.

b) **Secure Administration.** The TOE enables secure management of its security functions, including:

i) Administrator authentication with passwords

ii) Configurable password policies

iii) Role Based Access Control

iv) Access banners

v) Management of critical security functions and data

vi) Protection of cryptographic keys and passwords

c) **Trusted Update.** The TOE ensures the authenticity and integrity of software updates through digital signatures.

d) **System Monitoring.** The TOE generates logs of security relevant events. The TOE stores logs locally and is capable of sending log events to a remote audit server.

e) **Self-Test.** The TOE performs a suite of self-tests to ensure the correct operation and enforcement of its security functions.

f) **Cryptographic Operations.** The TOE implements a cryptographic module. Relevant Cryptographic Algorithm Validation Program (CAVP) certificates are shown in Table 4.

Table 4: CAVP Certificates

Module Name	Services	Certificates
Silver Peak EdgeConnect Cryptographic library, Crypto Library 2021 version 1.1	Provides cryptographic functions to support all SSH NTP, TLS and IPsec/IKE session operations	A5249 A5333
HPE BC-FJA (Bouncy Castle FIPS Java API), version 1.0.2	Performs cryptographic functions to support all TLS operations.	A4784 A5334
HPE Aruba Networking Orchestrator Cryptographic Library, Crypto Library 2024 version 1.0	Provides cryptographic functions to support all SSH and NTP operations	A5332

2.4 Physical Scope

15 The TOE boundary includes an EdgeConnect hardware appliance component and an Orchestrator virtual appliance component:

- The EdgeConnect Release 9.4.2 component consists of the EdgeConnect EC-XS Model 500210 running the ECOS 9.4.2 on Yocto 2.7.3 Warrior with Kernel 4.19.87.
- The Orchestrator Release 9.4.2 virtual appliance component consists of the Orchestrator 9.4.2 OVA software running on Rocky Linux 5.14.0.

16 The Orchestrator component is classified as a virtual network device (vND) corresponding to evaluated configuration Case 1 of Section 1.2 of CPP_ND_V2.2E. As such, the TOE boundary comprises the virtual machine (VM) software, but excludes the virtual system (hypervisor and hardware platform). The Orchestrator VM operates on the hypervisor VMware ESXi 7.0. For the evaluation, it was tested on a HPE ProLiant DL360 hardware platform. Table 5 shows the details on the evaluated configuration of the TOE.

Table 5: Evaluated Configuration

Component	HW Model	CPU	Software
EdgeConnect Release 9.4.2	EdgeConnect EC-XS Model 500210, P/N 201571	Intel® Atom C3558, 2.20 GHz, 4C/4T (Denverton)	ECOS 9.4.2 on Yocto 2.7.3 Warrior (4.19.87 Kernel)
Orchestrator Release 9.4.2	HPE ProLiant DL360	Intel Xeon-Gold 6242R (3.1GHz/20-core/205W) FIO Processor Kit for HPE ProLiant DL360 Gen10 (Cascade Lake)	Orchestrator 9.4.2 OVA on Rocky Linux 5.14.0 on VMware ESXi 7.0

17 The TOE is delivered via commercial courier.

2.4.1 Guidance Documents

18 The TOE includes the following guidance documents (PDF):

a) HPE Aruba Networking EdgeConnect SD-WAN Common Criteria Guidance, Orchestrator and ECOS Version 9.4.2, Version 1.3.4, December 2024

b) Using SD-WAN Orchestrator — 9.4.2, May 21, 2024 https://www.arubanetworks.com/techdocs/sdwan-PDFs/user/Orch_UserGuide_R942.pdf

c) CLI Reference, February 8, 2024 https://www.arubanetworks.com/techdocs/sdwan-PDFs/cli-ref/CLI-Reference_latest.pdf

2.4.2 Non-TOE Components

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

a) **Audit Server.** The TOE sends audit events to the remote syslog server.

b) **NTP Server.** Network Time Protocol server.

c) **Non-EdgeConnect VPN Peer.** Any VPN device which is not an EdgeConnect that may connect to the EdgeConnect TOE component.

d) **VMware hypervisors (ESX, ESXi, vSphere).** The Orchestrator operates on VMware ESXi 7.0.

e) **Management station.** Computer used to connect to the Orchestrator and EdgeConnect for management operations.

2.4.3 Functions not included in the TOE Evaluation

a) Rest API

3 Security Problem Definition

3.1 Threats

20 The following threats for this TOE are as defined in Sect. 4.1 of CPP_ND_V2.2E, Sect. 4.1 of MOD_CPP_FW_V1.4e, and Sect. 3.1 of MOD_VPNGW_V1.3.

Table 6: Threats

Identifier	Description
T.UNAUTHORIZED_ ADMINISTRATOR_ ACCESS	Threat agents may attempt to gain Administrator access to the Network Device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between Network Devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.
T.WEAK_ CRYPTOGRAPHY	Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.
T.UNTRUSTED_ COMMUNICATION_ CHANNELS	Threat agents may attempt to target Network Devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the Network Device itself.
T.WEAK_ AUTHENTICATION_ ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the Network Device itself could be compromised.
T.UPDATE_ COMPROMISE	Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.
T.UNDETECTED_ ACTIVITY	Threat agents may attempt to access, change, and/or modify the security functionality of the Network Device without Administrator awareness. This could result in the attacker finding an avenue (e.g.,

Identifier	Description
	misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_ FUNCTIONALITY_ COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.
T.PASSWORD_ CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other Network Devices.
T.SECURITY_ FUNCTIONALITY_ FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.
T.DATA_INTEGRITY (VPNGW)	Devices on a protected network may be exposed to threats presented by devices located outside the protected network that may attempt to modify the data without authorization. If known malicious external devices are able to communicate with devices on the protected network or if devices on the protected network can communicate with those external devices then the data contained in the communications may be susceptible to a loss of integrity.
T.NETWORK_ACCESS (VPNGW)	Devices located outside the protected network may seek to exercise services located on the protected network that are intended to only be accessed from inside the protected network or only accessed by entities using an authenticated path into the protected network. Devices located outside the protected network may, likewise, offer services that are inappropriate for access from within the protected network. From an ingress perspective, VPN gateways can be configured so that only those network servers intended for external consumption by entities operating on a trusted network (e.g., machines operating on a network where the peer VPN gateways are supporting the connection) are accessible and only via the intended ports. This serves to mitigate the potential for network entities outside a protected network to access network servers or services intended only for consumption or access inside a protected network. From an egress perspective, VPN gateways can be configured so that only specific external services (e.g., based on destination port) can be accessed from within a protected network, or moreover are accessed via an encrypted channel. For example, access to external mail services can be blocked to enforce corporate policies against accessing uncontrolled email servers, or, that access to the mail server must be done over an encrypted link.

Identifier	Description
T.NETWORK_ACCESS (FFW)	With knowledge of the services that are exported by machines on a subnet, an attacker may attempt to exploit those services by mounting attacks against those services.
T.NETWORK_DISCLOSURE (VPNGW)	Devices on a protected network may be exposed to threats presented by devices located outside the protected network, which may attempt to conduct unauthorized activities. If known malicious external devices are able to communicate with devices on the protected network, or if devices on the protected network can establish communications with those external devices (e.g., as a result of a phishing episode or by inadvertent responses to email messages), then those internal devices may be susceptible to the unauthorized disclosure of information. From an infiltration perspective, VPN gateways serve not only to limit access to only specific destination network addresses and ports within a protected network, but whether network traffic will be encrypted or transmitted in plaintext. With these limits, general network port scanning can be prevented from reaching protected networks or machines, and access to information on a protected network can be limited to that obtainable from specifically configured ports on identified network nodes (e.g., web pages from a designated corporate web server). Additionally, access can be limited to only specific source addresses and ports so that specific networks or network nodes can be blocked from accessing a protected network thereby further limiting the potential disclosure of information. From an exfiltration perspective, VPN gateways serve to limit how network nodes operating on a protected network can connect to and communicate with other networks limiting how and where they can disseminate information. Specific external networks can be blocked altogether, or egress could be limited to specific addresses or ports. Alternately, egress options available to network nodes on a protected network can be carefully managed in order to, for example, ensure that outgoing connections are encrypted to further mitigate inappropriate disclosure of data through packet sniffing.
T.NETWORK_DISCLOSURE (FFW)	An attacker may attempt to “map” a subnet to determine the machines that reside on the network, and obtaining the IP addresses of machines, as well as the services (ports) those machines are offering. This information could be used to mount attacks to those machines via the services that are exported.
T.NETWORK_MISUSE (VPNGW)	Devices located outside the protected network, while permitted to access particular public services offered inside the protected network, may attempt to conduct inappropriate activities while communicating with those allowed public services. Certain services offered from within a protected network may also represent a risk when accessed from outside the protected network. From an ingress perspective, it is generally assumed that entities operating on external networks are not bound by the use policies for a given protected network. Nonetheless, VPN gateways can log policy violations that might indicate violation of publicized usage statements for publicly available services.

Identifier	Description
	From an egress perspective, VPN gateways can be configured to help enforce and monitor protected network use policies. As explained in the other threats, a VPN gateway can serve to limit dissemination of data, access to external servers, and even disruption of services – all of these could be related to the use policies of a protected network and as such are subject in some regards to enforcement. Additionally, VPN gateways can be configured to log network usages that cross between protected and external networks and as a result can serve to identify potential usage policy violations.
T.NETWORK_MISUSE (FFW)	An attacker may attempt to use services that are exported by machines in a way that is unintended by a site's security policies. For example, an attacker might be able to use a service to "anonymize" the attacker's machine as they mount attacks against others.
T.REPLAY_ATTACK (VPNGW)	If an unauthorized individual successfully gains access to the system, the adversary may have the opportunity to conduct a "replay" attack. This method of attack allows the individual to capture packets traversing throughout the network and send the packets at a later time, possibly unknown by the intended receiver. Traffic is subject to replay if it meets the following conditions: • Cleartext: an attacker with the ability to view unencrypted traffic can identify an appropriate segment of the communications to replay as well in order to cause the desired outcome • No integrity: alongside cleartext traffic, an attacker can make arbitrary modifications to captured traffic and replay it to cause the desired outcome if the recipient has no means to detect these
T.MALICIOUS_TRAFFIC (FFW)	An attacker may attempt to send malformed packets to a machine in hopes of causing the network stack or services listening on UDP/TCP ports of the target machine to crash.

3.2 Assumptions

Table 7: Assumptions

Identifier	Description
A.PHYSICAL_PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.

Identifier	Description
A.LIMITED_ FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).
A.NO_THRU_ TRAFFIC_ PROTECTION	A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the NDcPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).
A.TRUSTED_ ADMINISTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
A.REGULAR_ UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_ CREDENTIALS_ SECURE	The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.
A.COMPONENTS_RUNNING	For distributed TOEs it is assumed that the availability of all TOE components is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. It is also assumed that in addition to the availability of all components it is also checked as appropriate that the audit functionality is running properly on all TOE components.
A.RESIDUAL_ INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

Identifier	Description
A.VS_TRUSTED_ADMINISTRATOR	The Security Administrators for the VS are assumed to be trusted and to act in the best interest of security for the organization. This includes not interfering with the correct operation of the device. The Network Device is not expected to be capable of defending against a malicious VS Administrator that actively works to bypass or compromise the security of the device.
A.VS_REGULAR_UPDATES	The VS software is assumed to be updated by the VS Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.VS_ISOLATON	For vNDs, it is assumed that the VS provides, and is configured to provide sufficient isolation between software running in VMs on the same physical platform. Furthermore, it is assumed that the VS adequately protects itself from software running inside VMs on the same physical platform.
A.VS_CORRECT_CONFIGURATION	For vNDs, it is assumed that the VS and VMs are correctly configured to support ND functionality implemented in VMs.
A.CONNECTIONS	It is assumed that the TOE is connected to distinct networks in a manner that ensures that the TOE's security policies will be enforced on all applicable network traffic flowing among the attached networks.

3.3 Organizational Security Policies

Table 8: Organizational Security Policies

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.

4 Security Objectives

21 The security objectives for the TOE and for the operational environment are taken from section 5 of the CPP_ND_V2.2E, section 5 of MOD_CPP_FW_V1.4e and section 4 of MOD_VPNGW_V1.3. Since the TOE has a virtual Network Device (vND) component, objectives for the vND component are included.

4.1 Security Objectives for the TOE

22 The NDcPP does not state security objectives for the TOE but argues that the security objectives for the TOE are trivially determined through the inverse of the

statement of threats presented in Sect. 4.1 of CPP_ND_V2.2E. Table 9 list the security objectives for the TOE defined in MOD_CPP_FW_V1.4e.

Table 9: Security Objectives for the TOE – MOD_CPP_FW_V1.4e

Identifier	Description
O.RESIDUAL_INFORMATION	The TOE shall implement measures to ensure that any previous information content of network packets sent through the TOE is made unavailable either upon deallocation of the memory area containing the network packet or upon allocation of a memory area for a newly arriving network packet or both.
O.STATEFUL_TRAFFIC_FILTERING	The TOE shall perform stateful traffic filtering on network packets that it processes. For this the TOE shall support the definition of stateful traffic filtering rules that allow to permit or drop network packets. The TOE shall support assignment of the stateful traffic filtering rules to each distinct network interface. The TOE shall support the processing of the applicable stateful traffic filtering rules in an administratively defined order. The TOE shall deny the flow of network packets if no matching stateful traffic filtering rule is identified. Depending on the implementation, the TOE might support the stateful traffic filtering of Dynamic Protocols (optional).

23 Table 10 lists the security objectives for the TOE defined in and MOD_VPNGW_V1.3.

Table 10: Security Objectives for the TOE – MOD_VPNGW_V1.3

Identifier	Description
O.ADDRESS_FILTERING	To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption or denial of services, and network-based reconnaissance, compliant TOE's will implement packet filtering capability. That capability will restrict the flow of network traffic between protected networks and other attached networks based on network addresses of the network nodes originating (source) or receiving (destination) applicable network traffic as well as on established connection information.
O.AUTHENTICATION	To further address the issues associated with unauthorized disclosure of information, a compliant TOE's authentication ability (IPSec) will allow a VPN peer to establish VPN connectivity with another VPN peer and ensure that any such connection attempt is both authenticated and authorized. VPN endpoints authenticate each other to ensure they are communicating with an authorized external IT entity.
O.CRYPTOGRAPHIC_FUNCTIONS	To address the issues associated with unauthorized disclosure of information, inappropriate access to services, misuse of services, disruption of services, and network-based reconnaissance, compliant TOE's will implement cryptographic capabilities. These capabilities are intended to maintain confidentiality and allow for detection and modification of data that is transmitted outside of the TOE.

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

4.2 Security Objectives for the Operational Environment

24 The following security objectives for the operational environment are defined in Section 5.1 of CPP_ND_V2.2E

Table 11: Security Objectives for the Operational Environment – CPP_ND_V2.2E

Identifier	Description
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

Identifier	Description
OE.NO_THRU_ TRAFFIC_ PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_ CREDENTIALS_ SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.COMPONENTS_R UNNING	For distributed TOEs, the Security Administrator ensures that the availability of every TOE component is checked as appropriate to reduce the risk of an undetected attack on (or failure of) one or more TOE components. The Security Administrator also ensures that it is checked as appropriate for every TOE component that the audit functionality is running properly.
OE.RESIDUAL_ INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

Identifier	Description
OE.VM_CONFIGURATION	For vNDs, the Security Administrator ensures that the VS and VMs are configured to • reduce the attack surface of VMs as much as possible while supporting ND functionality (e.g., remove unnecessary virtual hardware, turn off unused inter-VM communications mechanisms), and • correctly implement ND functionality (e.g., ensure virtual networking is properly configured to support network traffic, management channels, and audit reporting). The VS should be operated in a manner that reduces the likelihood that vND operations are adversely affected by virtualisation features such as cloning, save/restore, suspend/resume, and live migration. If possible, the VS should be configured to make use of features that leverage the VS's privileged position to provide additional security functionality. Such features could include malware detection through VM introspection, measured VM boot, or VM snapshot for forensic analysis.

25 There are no additional security objectives for the environment defined in MOD_CPP_FW_V1.4e but a clarification that OE.NO_THRU_TRAFFIC_PROTECTION only applies for the interfaces in the TOE that are defined by the Base-PP and not by the PP-Module.

26 The following security objectives for the operational environment are defined in MOD_VPNGW_V1.3.

Table 12: Security Objectives for the Operational Environment – MOD_VPNGW_V1.3

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

4.3 Security Objectives Rationale

27 Security objectives for the TOE and the security objectives for the operational environment are identical to those specified in Sect. 5 of CPP_ND_V2.2E, Sect. 5.3 of MOD_CPP_FW_V1.4e, and Sect. 4.3 of MOD_VPNGW_V1.3. The rationales are, therefore, also identical and are not reproduced here.

5 Security Requirements

5.1 Conventions

28 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 adding a string starting with "/" (e.g. "FCS_COP.1/Hash").

29 **Note:** Operations performed within the Security Target are denoted within brackets []. Operations shown without brackets are reproduced from the claimed PPs and modules.

5.2 Extended Components Definition

30 The Extended Components are defined in the claimed PPs and modules.

5.3 Functional Requirements

Table 13: Summary of SFRs

Requirement	Title	Component
FAU_GEN.1	Audit Data Generation	Orchestrator EdgeConnect
FAU_GEN.1/VPN	Audit Data Generation (VPN Gateway)	EdgeConnect
FAU_GEN.2	User Identity Association	Orchestrator EdgeConnect
FAU_GEN_EXT.1	Security Audit Generation	Orchestrator EdgeConnect
FAU_STG_EXT.1	Protected Audit Event Storage	Orchestrator EdgeConnect
FAU_STG_EXT.4	Protected Local Audit Event Storage for Distributed TOEs	Orchestrator EdgeConnect
FCO_CPC_EXT.1	Component Registration Channel Definition	Orchestrator EdgeConnect

Requirement	Title	Component
FCS_CKM.1	Cryptographic Key Generation	Orchestrator EdgeConnect
FCS_CKM.1/IKE	Cryptographic Key Generation (for IKE Peer Authentication)	EdgeConnect
FCS_CKM.2	Cryptographic Key Establishment	Orchestrator EdgeConnect
FCS_CKM.4	Cryptographic Key Destruction	Orchestrator EdgeConnect
FCS_COP.1/ DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)	Orchestrator EdgeConnect
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)	Orchestrator EdgeConnect
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)	Orchestrator EdgeConnect
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)	Orchestrator EdgeConnect
FCS_HTTPS_EXT.1	HTTPS Protocol	Orchestrator EdgeConnect
FCS_NTP_EXT.1	NTP Protocol	Orchestrator EdgeConnect
FCS_RBG_EXT.1	Random Bit Generation	Orchestrator EdgeConnect
FCS_IPSEC_EXT.1	IPSec Protocol	EdgeConnect
FCS_SSHS_EXT.1	SSH Server Protocol	Orchestrator EdgeConnect
FCS_TLSC_EXT.1	TLS Client Protocol Without Mutual Authentication	Orchestrator EdgeConnect
FCS_TLSS_EXT.1	TLS Server Protocol Without Mutual Authentication	Orchestrator EdgeConnect
FDP_RIP.2	Full Residual Information Protection	EdgeConnect

Requirement	Title	Component
FFW_RUL_EXT.1	Stateful Traffic Filtering	EdgeConnect
FIA_AFL.1	Authentication Failure Management	Orchestrator EdgeConnect
FIA_PMG_EXT.1	Password Management	Orchestrator EdgeConnect
FIA_UIA_EXT.1	User Identification and Authentication	Orchestrator EdgeConnect
FIA_UAU_EXT.2	Password-based Authentication Mechanism	Orchestrator EdgeConnect
FIA_UAU.7	Protected Authentication Feedback	Orchestrator EdgeConnect
FIA_X509_EXT.1/Rev	X.509 Certificate Validation	Orchestrator EdgeConnect
FIA_X509_EXT.1/ITT	X.509 Certificate Validation	Orchestrator EdgeConnect
FIA_X509_EXT.2	X.509 Certificate Authentication	Orchestrator EdgeConnect
FIA_X509_EXT.3	X.509 Certificate Requests	Orchestrator EdgeConnect
FMT_MOF.1/ManualUpdate	Management of Security Functions Behaviour	Orchestrator EdgeConnect
FMT_MTD.1/CoreData	Management of TSF Data	Orchestrator EdgeConnect
FMT_MTD.1/CryptoKeys	Management of TSF Data	Orchestrator EdgeConnect
FMT_SMF.1	Specification of Management Functions	Orchestrator EdgeConnect
FMT_SMF.1/FFW	Specification of Management Functions	Orchestrator EdgeConnect

Requirement	Title	Component
FMT_SMF.1/VPN	Specification of Management Functions	Orchestrator EdgeConnect
FMT_SMR.2	Restrictions on Security Roles	Orchestrator EdgeConnect
FPF_RUL_EXT.1	Packet Filtering Rules	EdgeConnect
FPT_ITT.1	Basic internal TSF data transfer protection	Orchestrator EdgeConnect
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)	Orchestrator EdgeConnect
FPT_APW_EXT.1	Protection of Administrator Passwords	Orchestrator EdgeConnect
FPT_FLS.1/SelfTest	Failure with Preservation of Secure State (Self-Test Failures)	Orchestrator EdgeConnect
FPT_TST_EXT.1	TSF Testing	Orchestrator EdgeConnect
FPT_TST_EXT.3	Self-Test with Defined Methods	Orchestrator EdgeConnect
FPT_TUD_EXT.1	Trusted Update	Orchestrator EdgeConnect
FPT_STM_EXT.1	Reliable Time Stamps	Orchestrator EdgeConnect
FTA_SSL_EXT.1	TSF-initiated Session Locking	Orchestrator EdgeConnect
FTA_SSL.3	TSF-initiated Termination	Orchestrator EdgeConnect
FTA_SSL.4	User-initiated Termination	Orchestrator EdgeConnect
FTA_TAB.1	Default TOE Access Banners	Orchestrator EdgeConnect

Requirement	Title	Component
FTP_ITC.1	Inter-TSF trusted channel	Orchestrator EdgeConnect
FTP_ITC.1/VPN	Inter-TSF Trusted Channel (VPN Communications)	EdgeConnect
FTP_TRP.1/Admin	Trusted Path	Orchestrator EdgeConnect

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 shutdown of the audit functions;

b) All auditable events for the not specified level of audit;

c) All administrative actions comprising:

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

d) *Specifically defined auditable events listed in **~~Table 2~~ Table 14**.*

Table 14: Audit Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.1/VPN	No events specified.	N/A
FAU_GEN.2	None.	None.
FAU_GEN_EXT.1	None.	None.
FAU_STG_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FAU_STG_EXT.4	None.	None.
FCO_CPC_EXT.1	• Enabling communications between a pair of components. • Disabling communications between a pair of components.	Identities of the endpoint pairs enabled or disabled.
FCS_CKM.1	None.	None.
FCS_CKM.1/IKE	No events specified	N/A
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA.	Reason for failure
FCS_NTP_EXT.1	• Configuration of a new time server • Removal of configured time server	Identity if new/removed time server
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure

Requirement	Auditable Events	Additional Audit Record Contents
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FDP_RIP.2	None.	None.
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
FPF_RUL_EXT.1	Application of rules configured with the 'log' operation	• Source and destination addresses • Source and destination ports • Transport layer protocol
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store •	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store •

Requirement	Auditable Events	Additional Audit Record Contents
FIA_X509_EXT.1/ITT	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store •	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store •
FIA_X509_EXT.2	None	None
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Functions	None.	None.
FMT_MOF.1/Services	None	None
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMF.1/FFW	All management activities of TSF data (including creation, modification and deletion of firewall rules).	None.
FMT_SMF.1/VPN	All administrative actions	No additional information.
FMT_SMR.2	None.	None.
FPT_ITT.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FPT_SKP_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FPT_APW_EXT.1	None.	None.
FPT_FLS.1/SelfTest	No events specified.	N/A
FPT_TST_EXT.1	None.	None.
FPT_TST_EXT.3	No events specified.	N/A
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_ITC.1/VPN	• Initiation of the trusted channel • Termination of the trusted channel	No additional information

Requirement	Auditable Events	Additional Audit Record Contents
	Failure of a trusted channel function	Identification of the initiator and target of failed trusted channel establishment attempt
FTP_TRP.1/Admin	• Initiation of the trusted path. • Termination of the trusted path. • Failure of the trusted path functions.	None.

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of* ***~~Table 2~~ Table 14***.

FAU_GEN.1/VPN Audit Data Generation (VPN Gateway)

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

a) Start-up and shutdown of the audit functions

b) Indication that TSF self-test was completed

c) Failure of self-test

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

e) [*auditable events defined in the Audit Events table*].

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*additional information defined in the Audit Events table for each auditable event, where applicable*].

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_GEN_EXT.1 Security Audit Generation

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

FAU_STG_EXT.1 Protected Audit Event Storage

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

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

- The TOE shall be a distributed TOE that stores audit data on the following TOE components: [*Orchestrator, EdgeConnect*],

]

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

FAU_STG_EXT.4 Protected Local Audit Event Storage for Distributed TOEs

FAU_STG_EXT.4.1 The TSF of each TOE component which stores security audit data locally shall perform the following actions when the local storage space for audit data is full: [*Orchestrator*: overwrite previous audit records according to the following rule: [*oldest records will be overwritten*], *EdgeConnect:* overwrite previous audit records according to the following rule: [*oldest records will be overwritten*]].

5.3.2 Communication (FCO)

FCO_CPC_EXT.1 Component Registration Channel Definition

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

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

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

]

for at least TSF data.

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

5.3.3 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

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

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

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

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

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

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

] and [

- FFC Schemes using “safe-prime” groups that meet the following:‘NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and [RFC 3526],

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

FCS_CKM.2 Cryptographic Key Establishment

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

- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447,

"Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1";

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";
- FFC Schemes using "safe-prime" groups that meet the following: 'NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [groups listed in RFC 3526, groups listed in RFC 7919];

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

Application note: This SFR was changed by TD0580 and TD0581.

FCS_CKM.4 Cryptographic Key Destruction

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

- *For plaintext keys in volatile storage, the destruction shall be executed by a [single overwrite consisting of [zeroes]];*
- *For plaintext keys in non-volatile storage, the destruction shall be executed by the invocation of an interface provided by a part of the TSF that [*
    - *instructs a part of the TSF to destroy the abstraction that represents the key*

] that meets the following: *No Standard*.

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

FCS_COP.1.1/DataEncryption The TSF shall perform encryption/decryption in accordance with a specified cryptographic algorithm AES used in ***[CBC, GCM] and [CTR]*** mode and cryptographic key sizes [128 bits, 256 bits] that meet the following: AES as specified in ISO 18033-3, ***[CBC as specified in ISO 10116, GCM as specified in ISO 19772], and [CTR as specified in ISO 10116]***.

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

FCS_COP.1.1/SigGen The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [*2048 bits, 3072 bits and 4096 bits*],
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [*256 bits, 384 bits and 521 bits*],

] that meet the following: [

- For RSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,
- For ECDSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4]

FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

FCS_COP.1.1/Hash The TSF shall perform *cryptographic hashing services* in accordance with a specified cryptographic algorithm [SHA-1, SHA-256, SHA-384, SHA-512] ~~and cryptographic key sizes [assignment: cryptographic key sizes]~~ and **message digest sizes [160, 256, 384, 512] bits** that meet the following: *ISO/IEC 10118-3:2004*.

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

FCS_COP.1.1/KeyedHash The TSF shall perform *keyed-hash message authentication* in accordance with a specified cryptographic algorithm [HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] and cryptographic key sizes *[160, 256, 384, 512]* **and message digest sizes [160, 256, 384, 512] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”*.

FCS_HTTPS_EXT.1 HTTPS Protocol

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

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [not require client authentication] if the peer certificate is deemed invalid.

FCS_IPSEC_EXT.1 IPsec Protocol

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

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

FCS_IPSEC_EXT.1.3 The TSF shall implement [tunnel mode].

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

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [

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

].

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

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [

- IKEv2 SA lifetimes can be configured by a Security Administrator based on [
  - length of time, where the time values can be configured within [*1 minute to 24*] hours

  ]

].

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [

- IKEv2 Child SA lifetimes can be configured by a Security Administrator based on [
  - number of bytes;
  - length of time, where the time values can be configured within [*1 minute to 8*] hours;

  ]

].

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least [*224 (for DH Group 14), 256 (for DH Groups 15 and 19), 350 (for DH Group 17), 384 (for DH Groups 18 and 20), and 512 (for DH Group 21)*] bits.

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

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

].

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

- **19 (256-bit Random ECP), 20 (384-bit Random ECP) according to RFC 5114 and**

[

- [14 (2048-bit MODP), 15 (3072-bit MODP), 16 (4096-bit MODP),17 (6144-bit MODP), 18 (8192-bit MODP)] according to RFC 3526
- [21 (521-bit Random ECP)] according to RFC 5114

].

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

FCS_IPSEC_EXT.1.13 The TSF shall ensure that **[IKEv2]** protocols perform peer authentication using [RSA, ECDSA] that use X.509v3 certificates that conform to RFC 4945 and [Pre-shared Keys that conform to RFC 8784].

Application Note:

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

Application note: This SFR was changed by MOD_VPNGW_V1.3.

FCS_NTP_EXT.1 NTP Protocol

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

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

- Authentication using [SHA384] as the message digest algorithm(s);

].

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

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

FCS_RBG_EXT.1 Random Bit Generation

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

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

FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFC(s) 4251, 4252, 4253, 4254, [4256, 4344, 5656, 6668, 8268, 8308 section 3.1, 8332].

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

Application note: This SFR was changed by TD0631.

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*256 kilo*]bytes in an SSH transport connection are dropped.

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

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521] as its public key algorithm(s) and rejects all other public key algorithms.

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

FCS_SSHS_EXT.1.7 The TSF shall ensure that [diffie-hellman-group14-sha1, ecdh-sha2-nistp256] and [diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, ecdh-sha2-nistp384, ecdh-sha2-nistp521] are the only allowed key exchange methods used for the SSH protocol.

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

FCS_TLSC_EXT.1 TLS Client Protocol Without Mutual Authentication

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

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492

- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

] and no other ciphersuites.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches [the reference identifier per RFC 6125 section 6, IPv4 address in CN or SAN, IPv4 address in SAN].

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

- Not implement any administrator override mechanism

].

FCS_TLSC_EXT.1.4 The TSF shall [present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups: [secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096] and no other curves/groups] in the Client Hello.

FCS_TLSS_EXT.1 TLS Server Protocol Without Mutual Authentication

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

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5288
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5288
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

] and no other ciphersuites.

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [TLS 1.1].

FCS_TLSS_EXT.1.3 The TSF shall perform key establishment for TLS using [Diffie-Hellman parameters with size [2048 bits], Diffie-Hellman groups ffdhe2048, ffdhe3072, ffdhe4096], ECDHE curves [secp256r1, secp384r1, secp521r1] and no other curves]].

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

Application Note: The Orchestrator supports no TLS session resumption, and the EdgeConnect supports TLS session resumption using session tickets,

5.3.4 Full Residual Information 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.

5.3.5 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 allow the definition of stateful traffic filtering rules using the following network protocol fields:

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

and distinct interface.

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

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

FFW_RUL_EXT.1.5 The TSF shall:

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

1. TCP: source and destination addresses, source and destination ports, sequence number, Flags;
2. UDP: source and destination addresses, source and destination ports;
3. ['ICMP: source and destination addresses, type, [code]'].

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

FFW_RUL_EXT.1.6 The TSF shall enforce the following default stateful traffic filtering rules on all network traffic:

a) The TSF shall drop and be capable of [counting, logging] packets which are invalid fragments;

b) The TSF shall drop and be capable of [counting, logging] fragmented packets which cannot be re-assembled completely;

c) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a broadcast network;

d) The TSF shall drop and be capable of logging packets where the source address of the network packet is defined as being on a multicast network;

e) The TSF shall drop and be capable of logging network packets where the source address of the network packet is defined as being a loopback address;

f) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address "reserved for future use" (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;

g) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is defined as an "unspecified address" or an address "reserved for future definition and use" (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;

h) The TSF shall drop and be capable of logging network packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified; and

i) [no other rules].

FFW_RUL_EXT.1.7 The TSF shall be capable of dropping and logging according to the following rules:

a) The TSF shall drop and be capable of logging network packets where the source address of the network packet is equal to the

address of the network interface where the network packet was received;

b) The TSF shall drop and be capable of logging network packets where the source or destination address of the network packet is a link-local address;

c) The TSF shall drop and be capable of logging network packets where the source address of the network packet does not belong to the networks associated with the network interface where the network packet was received.

FFW_RUL_EXT.1.8 The TSF shall process the applicable stateful traffic filtering rules in an administratively defined order.

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

FFW_RUL_EXT.1.10 The TSF shall be capable of limiting an administratively defined number of half-open TCP connections. In the event that the configured limit is reached, new connection attempts shall be dropped and the drop event shall be [counted].

5.3.6 Identification and Authentication (FIA)

FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1-1000*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

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

FIA_PMG_EXT.1 Password Management

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”];

b) Minimum password length shall be configurable to between [*8*] and [*64] characters.*

FIA_PSK_EXT.1 Pre-Shared Key Composition

FIA_PSK_EXT.1.1 The TSF shall be able to use pre-shared keys for IPsec and [IKEv2]

Application Note: Pre-shared key authentication is supported only in the Orchestrator.

FIA_PSK_EXT.1.2 The TSF shall be able to accept the following as pre-shared keys: [generated bit-based] keys.

FIA_PSK_EXT.2 Generated Pre-Shared Keys

FIA_PSK_EXT.2.1 The TSF shall be able to [

- accept externally generated pre-shared keys

]

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 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local [password-based] authentication mechanism to perform local 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**.

FIA_X509_EXT.1/Rev X.509 Certificate Validation

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

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of three certificates**.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 6960].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-

kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.

- Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
- Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
- OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

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

FIA_X509_EXT.1/ITT X.509 Certificate Validation

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of two certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 6960]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

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

FIA_X509_EXT.2 X.509 Certificate Authentication

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

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

Application note: This SFR was changed by MOD_VPNGW_V1.3.

FIA_X509_EXT.3 X.509 Certificate Requests

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

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

5.3.7 Security Management (FMT)

FMT_MOF.1/ManualUpdate Management of security functions behaviour

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

FMT_MOF.1/Functions Management of Security Functions Behaviour

FMT_MOF.1.1/Functions The TSF shall restrict the ability to [modify the behaviour of] the functions [audit functionality when Local Audit Storage Space is full] to *Security Administrators.*

FMT_MOF.1/Services Management of Security Functions Behaviour

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

FMT_MTD.1/CoreData Management of TSF Data

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

FMT_MTD.1/CryptoKeys Management of TSF Data

FMT_MTD.1.1/CryptoKeys The TSF shall restrict the ability to [[*manage*]] the [*cryptographic keys **and certificates used for VPN operation***] to [*Security Administrators*].

FMT_SMF.1 Specification of Management Functions

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- [
  - Ability to start and stop services;
  - Ability to configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full);
  - Ability to manage the cryptographic keys;
  - Ability to configure the cryptographic functionality;
  - Ability to configure the lifetime for IPsec SAs;
  - Ability to set the time which is used for time-stamps;
  - Ability to configure NTP;
  - Ability to configure the reference identifier for the peer;
  - Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;
  - Ability to import X.509v3 certificates to the TOE's trust store;
  - Ability to manage the trusted public keys database;

  ]

FMT_SMF.1/FFW Specification of Management Functions

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

- *Ability to configure firewall rules;*

FMT_SMF.1/VPN Specification of Management Functions

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

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

[

- No other capabilities

]].

FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- *Security Administrator*.

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

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

are satisfied.

5.3.8 Packet Filtering (FPF)

FPF_RUL_EXT.1 Packet Filtering Rules

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 allow the definition of packet filtering rules using the following network protocols and protocol fields: [

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

].

FPF_RUL_EXT.1.3	The TSF shall allow the following operations to be associated with packet filtering rules: permit and drop with the capability to log the operation.
FPF_RUL_EXT.1.4	The TSF shall allow the packet filtering rules to be assigned to each distinct network interface.
FPF_RUL_EXT.1.5	The TSF shall process the applicable packet filtering rules (as determined in accordance with FPF_RUL_EXT.1.4) in the following order: [*Administrator defined*].
FPF_RUL_EXT.1.6	The TSF shall drop traffic if a matching rule is not identified.

5.3.9 Protection of the TSF (FPT)

FPT_ITT.1 Basic internal TSF data transfer protection

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

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

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

FPT_APW_EXT.1 Protection of Administrator Passwords

FPT_APW_EXT.1.1	The TSF shall store administrative passwords in non-plaintext form.
FPT_APW_EXT.1.2	The TSF shall prevent the reading of plaintext administrative passwords.

FPT_FLS.1/SelfTest Failure with Preservation of Secure State (Self-Test Failures)

FPT_FLS.1.1/SelfTest	The TSF shall **shut down** when the following types of failures occur: [*failure of the power-on self-tests, failure of integrity check of the TSF executable image, failure of noise source health tests*].

FPT_TST_EXT.1 TSF testing

FPT_TST_EXT.1.1	The TSF shall run a suite of the following self-tests [during initial start-up (on power on)] to demonstrate the correct operation of the TSF: [• *BIOS memory test* • *System integrity check* • *Cryptographic self-tests*].

FPT_TST_EXT.3 Self-Test with Defined Methods

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

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

FPT_TUD_EXT.1 Trusted update

FPT_TUD_EXT.1.1 The TSF shall provide *Security Administrators* the ability to query the currently executing version of the TOE firmware/software and [the most recently installed version of the TOE firmware/software, no other TOE firmware/software version].

FPT_TUD_EXT.1.2 The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and [support automatic checking for updates, no other update mechanism].

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [digital signature] prior to installing those updates.

FPT_STM_EXT.1 Reliable Time Stamps

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

FPT_STM_EXT.1.2 The TSF shall [allow the Security Administrator to set the time, synchronize time with an NTP server].

5.3.10 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 TSF-initiated Termination

FTA_SSL.3.1 The TSF shall terminate **a remote** interactive session after a *Security Administrator-configurable time interval of session inactivity*.

FTA_SSL.4 User-initiated Termination

FTA_SSL.4.1 Refinement: 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 Before establishing **an administrative user** session the TSF shall display **a Security Administrator-specified** advisory **notice and consent** warning message regarding use of the TOE.

5.3.11 Trusted path/channels (FTP)

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall **be capable of using [TLS] to provide** a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [no other capabilities]** 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 [*communications with an audit server*].

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

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

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

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

FTP_TRP.1 /Admin Trusted Path

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

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

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

5.4 Security Assurance Requirements

31 The TOE security assurance requirements are summarized in Table 15.

Table 15: Assurance Requirements

Assurance Class	Components	Description
Security Target Evaluation (ASE)	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST Introduction
	ASE_OBJ.1	Security Objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE summary specification
Development (ADV)	ADV_FSP.1	Basic functional specification
Guidance Documents (AGD)	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative user guidance
Life-Cycle Support (ALC)	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests (ATE)	ATE_IND.1	Independent testing - conformance
Vulnerability Analysis (AVA)	AVA_VAN.1	Vulnerability survey

32 In accordance with section 7.1 of the CPP_ND_V2.2E, the following refinement is made to ASE:

a) **ASE_TSS.1.1C Refinement:** The TOE summary specification shall describe how the TOE meets each SFR. **In the case of entropy analysis, the TSS is used in conjunction with required supplementary information on Entropy.**

5.5 Security Requirements Rationale

33 The Security Requirements for the TOE are taken from CPP_ND_V2.2E, MOD_CPP_FW_V1.4e, and MOD_VPNFW. Only operations allowed in them are implemented. Therefore, the security requirement rationales in CPP_ND_V2.2E, MOD_CPP_FW_V1.4e, and MOD_VPNFW are directly applicable and are not repeated here.

6 TOE Summary Specification

34 The following describes how the TOE fulfils each SFR included in section 5.3.

6.1 Security Audit

6.1.1 FAU_GEN.1

35 The TOE generates the audit records specified at FAU_GEN.1 containing fields that include the timestamp, IP address (if applicable), action, user (if applicable) and a contextual message indicating success or failure of the action.

36 The following information is logged as a result of the Security Administrator generating/importing or deleting cryptographic keys:

a) **Generate SSH key-pair**. Action and key reference.

b) **Import of user public keys**. Action and key reference

c) **Generate X509 certificates requests**. Action and cert reference.

d) **Import of certificates**. Action and cert reference.

Table 16 identifies the TOE components that generate the auditable events defined in FAU_GEN.1.1.

Table 16: Audit Events

Requirement	Auditable Events	TOE Component
FAU_GEN.1	Start-up and shutdown of the audit functions	All
	Administrative login and logout (Name of user account shall be logged if individual user accounts are required for Administrators)	All
	Changes to TSF data related to configuration changes (In addition to the information that a change occurred it shall be logged what has been changed)	All
	Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged)	All
	Resetting passwords (name of related user account shall be logged)	All
FAU_GEN.1/VPN	Indication that TSF self-test was completed	EdgeConnect
	Failure of self-test	EdgeConnect

Requirement	Auditable Events	TOE Component
FCO_CPC_EXT.1	Enabling communications between a pair of components. Disabling communications between a pair of components. (Identities of the endpoints pairs enabled or disabled.)	All
FCS_IPSEC_EXT.1	Failure to establish an IPsec SA	EdgeConnect
FCS_NTP_EXT.1	Configuration of a new time server Removal of configured time server	All
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	All
FCS_SSHS_EXT.1	Failure to establish an SSH session	All
FCS_TLSC_EXT.1	Failure to establish a TLS Session	All
FCS_TLSS_EXT.1	Failure to establish a TLS Session	All
FFW_RUL_EXT.1	Application of rules configured with the 'log' operation	EdgeConnect
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	All
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	All
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	All
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate Any addition, replacement or removal of trust anchors in the TOE's trust store	All
FIA_X509_EXT.1/ITT	Unsuccessful attempt to validate a certificate Any addition, replacement or removal of trust anchors in the TOE's trust store	EdgeConnect
FMT_MOF.1/ ManualUpdate	Any attempt to initiate a manual update	All
FMT_SMF.1	All management activities of TSF data.	All

Requirement	Auditable Events	TOE Component
FMT_SMF.1/FFW	All management activities of TSF data (including creation, modification and deletion of firewall rules).	All
FMT_SMF.1/VPN	All administrative actions	All
FPT_ITT.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	All
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	All
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	All
FTA_SSL_EXT.1	The termination of a local session by the session locking mechanism.	All
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	All
FTA_SSL.4	The termination of an interactive session.	All
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	All
FTP_ITC.1/VPN	Initiation of the trusted channel Termination of the trusted channel Failure of the trusted channel functions	EdgeConnect
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	All

6.1.2 FAU_GEN.1/VPN

37 Each TOE component generates the audit records specified at FAU_GEN.1/VPN containing fields that include the timestamp, IP address (if applicable), action, user (if applicable) and a contextual message indicating success or failure of the action. The

audit mechanism used for VPN functionality is identical to the one used for all other TOE Security Functions.

6.1.3 FAU_GEN.2

38 The TOE includes the user identity in audit events resulting from actions of identified users.

6.1.4 FAU_GEN_EXT.1

39 Audit records are generated for each TOE component and include the subset of security relevant audit events which can occur on the TOE component.

6.1.5 FAU_STG_EXT.1

40 The audit records are securely sent to a remote audit server in the operational environment using TLS (see FCS_TLSC_EXT.1). This prevents the audit records from unauthorized viewing and modification during transmission. Both TOE components transmit audit data to the remote audit server in real time using TLS.

41 The TOE logs all events related to startup/shutdown, external communications, user authentication, and user management (user creation/deletion, password changes, role changes) and administrative commands in the audit log.

42 The TOE is a distributed TOE with both components storing audit data locally in rotating log files as follows:

43 **/var/log log files.** The Orchestrator creates a new log file every day or when a file reaches 1024MB in size, keeping at most 30 files or removing files every 30 days. The EdgeConnect has a configurable amount between 1MB and 50MB of data is kept in each of the log files before they are rotated. A configurable number between 1 and 100 of previous log files are kept of each log file and one live log. When the maximum storage space for log data is reached, the TOE overwrites previous audit records by removing the oldest log file and creating a new one.

44 Only authorized administrators may view audit records and no capability to modify the audit records is provided.

6.1.6 FAU_STG_EXT.4

45 Each TOE component performs the following actions when the local storage space for audit data is full:

46 Orchestrator:

47 **/var/log log files.** Creates a new log file every day or when a file reaches 1024MB in size, keeping at most 30 files or removing files every 30 days. Oldest logs are removed first.

48 EdgeConnect

49 **/var/log log files.** Configurable amount between 1MB and 50MB of data is kept in each of the log files before they are rotated. Configurable number between 1 and 100 previous log files are kept of each log file and one live log. Oldest logs are removed first.

6.2 Communication

6.2.1 FCO_CPC_EXT.1

50 The TOE requires a Security Administrator to enable communications between any pair of TOE components before such communication can take place. Communications is enabled via the HTTPS Web GUI by configuring the Orchestrator address on the EdgeConnect component. Once the EdgeConnect component is discovered by the Orchestrator, the EdgeConnect must be approved via the Orchestrator into the list of devices. The Orchestrator approval process involves assigning the component a name and a group that is used to identify the device in the approved device list.

51 The TOE implements a registration process in which components establish and use a communications channel that meets the secure channel requirements in FPT_ITT.1.

52 The TOE enables a Security Administrator to disable communications between any pair of TOE components. Communications is disabled by removing the Orchestrator from the EdgeConnect configuration and removing the EdgeConnect from the Orchestrator device list.

6.3 Cryptographic Support

53 The TOE includes the following FIPS 140-2 Level 2 certified cryptographic modules which provide supporting cryptographic functions: Silver Peak EdgeConnect Cryptographic library, and HPE BC-FJA (Bouncy Castle FIPS Java API).

54 The Silver Peak EdgeConnect Cryptographic library and HPE BC-FJA (Bouncy Castle FIPS Java API) are used for SSH/HTTPS/TLS cryptographic functions, the Silver Peak EdgeConnect Cryptographic library and is used for IPsec/IKE session cryptography. All modules implement the low-level cryptographic function in support of the protocols and run self-tests. The CAVP certificates below are defined in Table 4.

Table 17: SFR to CAVP Mapping

Library Implemented	Cryptographic and Applicable SFRs	Function, Usage, Algorithm, Mode, Key Size	CAVP Reference
Orchestrator: HPE BC-FJA (Bouncy Castle FIPS Java API) 1.0.2 HPE Aruba Networking Orchestrator Cryptographic Library, Crypto Library 2024 version 1.0 EdgeConnect: Silver Peak EdgeConnect Cryptographic library 1.1	FCS_CKM.1 FCS_SSHS_EXT.1 FCS_TLSS_EXT.1 FCS_TLSC_EXT.1	RSA KeyGen (FIPS Pub 186-4) (2048-bit, 3072-bit) ECDSA KeyGen (FIPS Pub 186-4) (P-256, P-384, P-521) FFC Safe Prime Groups (NIST SP 800-56A Rev. 3, RFC 3526)	Orchestrator: A4784 A5332 A5334 EdgeConnect: A5249 A5333
EdgeConnect:	FCS_CKM.1/IKE	RSA KeyGen (FIPS Pub 186-4)	EdgeConnect:

Silver Peak EdgeConnect Cryptographic library 1.1	FCS_IPSEC_EXT.1 FIA_X509_EXT.3	(2048-bit, 3072-bit) ECDSA KeyGen (FIPS Pub 186-4) (P-256, P-384, P-521) FFC Safe Prime Groups (NIST SP 800-56A Rev. 3, RFC 3526)	A5249 A5333
Orchestrator: HPE BC-FJA (Bouncy Castle FIPS Java API) 1.0.2 HPE Aruba Networking Orchestrator Cryptographic Library, Crypto Library 2024 version 1.0 EdgeConnect: Silver Peak EdgeConnect Cryptographic library 1.1	FCS_CKM.2 FCS_SSHS_EXT.1 FCS_IPSEC_EXT.1 FCS_TLSS_EXT.1 FCS_TLSC_EXT.1	RSA-based key establishment schemes (RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447 “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1”) Elliptic Curve-based Schemes (NIST SP 800-56A Rev 3) KAS-ECC-SSC (ECDH) (P-256, P-384, P-521) KAS-FFC-SSC (p=2048, q=256) (p=3072, q=384) (p=4096, q=512) FFC Safe Prime Groups (NIST SP 800-56A Rev 3 and Groups Listed in RFC 3526) DH Groups: SSH: •Group 14 per RFC 3526 section 3 •Group 16 per RFC 3526 section 5 •Group 18 per RFC 3526 section 7 IPSEC •Group 14 per RFC 3526 section 3 •Group 15 per RFC 3526 section 4 •Group 16 per RFC 3526 section 5 •Group 17 per RFC 3526 section 6 •Group 18 per RFC 3526 section 7f	Orchestrator: A4784 A5332 EdgeConnect: A5249 A5333
Orchestrator: HPE BC-FJA (Bouncy Castle FIPS Java API) 1.0.2 HPE Aruba Networking Orchestrator Cryptographic Library, Crypto Library 2024 version 1.0 EdgeConnect: Silver Peak EdgeConnect Cryptographic library 1.1	FCS_COP.1/DataEncryption FCS_SSHS_EXT.1 FCS_IPSEC_EXT.1 FCS_TLSS_EXT.1 FCS_TLSC_EXT.1	AES CBC (128 and 256 bits) AES GCM (128 and 256 bits) AES CTR (128 and 256 bits)	Orchestrator: A4784 A5332 EdgeConnect: A5249
Orchestrator: HPE BC-FJA (Bouncy Castle FIPS Java API) 1.0.2 HPE Aruba Networking	FCS_COP.1/SigGen FCS_SSHS_EXT.1 FCS_IPSEC_EXT.1 FCS_TLSS_EXT.1 FCS_TLSC_EXT.1	RSA SigGen (FIPS 186-4) (modulus 2048, 3072, 4096 bits) RSA SigVer (FIPS 186-4) (modulus 2048, 3072, 4096 bits) ECDSA SigGen (FIPS 186-4) (256 bits, 384 bits, 521 bits) ECDSA SigVer (FIPS 186-4)	Orchestrator: A4784 A5332 EdgeConnect: A5249 A5333

Orchestrator Cryptographic Library, Crypto Library 2024 version 1.0 EdgeConnect: Silver Peak EdgeConnect Cryptographic library 1.1		(256 bits, 384 bits, 521 bits)	
Orchestrator: HPE BC-FJA (Bouncy Castle FIPS Java API) 1.0.2 HPE Aruba Networking Orchestrator Cryptographic Library, Crypto Library 2024 version 1.0 EdgeConnect: Silver Peak EdgeConnect Cryptographic library 1.1	FCS_COP.1/Hash FCS_SSHS_EXT.1 FCS_IPSEC_EXT.1 FCS_TLSS_EXT.1 FCS_TLSC_EXT.1 FCS_NTP_EXT.1 FPT_TUD_EXT.1	SHA-1 SHA-256 SHA-384 SHA-512 (160, 256, 384 and 512 bits respectively)	Orchestrator: A4784 A5332 EdgeConnect: A5249
Orchestrator: HPE BC-FJA (Bouncy Castle FIPS Java API) 1.0.2 HPE Aruba Networking Orchestrator Cryptographic Library, Crypto Library 2024 version 1.0 EdgeConnect: Silver Peak EdgeConnect Cryptographic library 1.1	FCS_COP.1/KeyedHash FCS_SSHS_EXT.1 FCS_IPSEC_EXT.1 FCS_TLSS_EXT.1 FCS_TLSC_EXT.1	HMAC-SHA-1 HMAC-SHA-256 HMAC-SHA-384 HMAC-SHA-512 (160, 256, 384 and 512 bits respectively)	Orchestrator: A4784 A5332 EdgeConnect: A5249
Orchestrator: HPE BC-FJA (Bouncy Castle FIPS Java API) 1.0.2 HPE Aruba Networking Orchestrator Cryptographic Library, Crypto Library 2024 version 1.0	FCS_RBG_EXT.1 FCS_IPSEC_EXT.1 FCS_SSHS_EXT.1 FCS_TLSS_EXT.1 FCS_TLSC_EXT.1	CTR_DRBG (AES) (256 bits)	Orchestrator: A4784 A5332 EdgeConnect: A5249

EdgeConnect: Silver Peak EdgeConnect Cryptographic library 1.1			

6.3.1 FCS_CKM.1

55 The TOE supports key generation for the following asymmetric schemes:

a) **RSA 2048/3072.** Used in SSH and TLS authentication.

b) **ECC P-256/P-384/P-521.** Used in SSH and TLS authentication and key exchange.

c) **FFC Safe Primes.** Used in SSH and TLS key exchange.

6.3.2 FCS_CKM.1/IKE

56 The TOE supports key generation used for IKE peer authentication using the following asymmetric schemes:

a) **RSA schemes:** Used in authentication.

b) **ECC P-256/P-384/P-521.** Used in authentication and key exchange.

c) **FFC Safe Primes.** Used in key exchange.

57 The TOE supports key sizes equivalent to, or greater than, a symmetric key strength of 112 bits.

58 FIPS 186-4, Appendix B.3 The TOE implements all “shall” and “should” statements and does not implement any ‘"shall not" " or "should not" statements.

59 Details of “should” statements:

a) Pg. 64 & 65 – If an error is encountered during the generation process invalid values are returned.

60 FIPS 186-4, Appendix B.4 The TOE implements all “shall” and “should” statements and does not implement any ‘"shall not" " or "should not" statements.

61 Details of “should” statements:

a) Pg. 63 – If an error is encountered during the generation process invalid values are returned.

6.3.3 FCS_CKM.2

62 The TOE supports the following key establishment schemes:

a) **RSA schemes.** Used in TLS key exchange. TOE is the client.

b) **ECC schemes.** Used in SSH, TLS and IPSEC key exchange. TOE is server in SSH, and both client and server for TLS and IPSEC.

c) **FFC schemes using safe primes.** Used in SSH, TLS and IPSEC key exchange. TOE is server in SSH, and both client and server for TLS and IPSEC. The TOE meets RFC 3526 Section 3.

63 Table 18 below identifies the scheme being used by each service.

Table 18: Key Agreement Mapping

Scheme	SFR	Service	Component
RSA	FCS_TLSC_EXT.1	Audit Server	Orchestrator EdgeConnect
	FCS_TLSC_EXT.1	ITT	EdgeConnect
ECC	FCS_SSHS_EXT.1	Administration	Orchestrator EdgeConnect
	FCS_TLSS_EXT.1	Administration	Orchestrator EdgeConnect
	FCS_TLSS_EXT.1	ITT	Orchestrator
	FCS_TLSC_EXT.1	Audit Server	Orchestrator EdgeConnect
	FCS_TLSC_EXT.1	ITT	EdgeConnect
	FCS_IPSEC_EXT.1	VPN	EdgeConnect
FFC Safe Primes	FCS_SSHS_EXT.1	Administration	Orchestrator EdgeConnect
	FCS_TLSS_EXT.1	Administration	Orchestrator EdgeConnect
	FCS_TLSS_EXT.1	ITT	Orchestrator
	FCS_TLSC_EXT.1	Audit Server	Orchestrator EdgeConnect
	FCS_TLSC_EXT.1	ITT	EdgeConnect
	FCS_IPSEC_EXT.1	VPN	EdgeConnect

6.3.4 FCS_CKM.4

64 Table 21 shows the origin, storage location and destruction details for cryptographic keys. Unless otherwise stated, the keys are generated by the TOE.

6.3.5 FCS_COP.1/DataEncryption

65 The TOE provides symmetric encryption and decryption capabilities using 128 and 256 bit AES in CBC, CTR and GCM mode. AES is implemented in SSH, TLS and IPsec.

66 The relevant NIST CAVP certificate numbers are listed Table 4.

6.3.6 FCS_COP.1/SigGen

67 The TOE provides cryptographic signature generation and verification services using:

a) RSA Signature Algorithm with key sizes of 2048 bits, 3072 bits and 4096 bits.

b) ECDSA Signature Algorithm with key sizes of 256, 384 and 521 bits.

68 The RSA and ECDSA signature generation services are used in the SSH, TLS and IPSEC protocols.

69 The RSA signature verification services are used for the SSH, TLS and IPSEC protocols and TOE firmware integrity checks.

70 The ECDSA signature verification services are used for Trusted Updates, SSH, TLS and IPSEC protocols.

71 The relevant NIST CAVP certificate numbers are listed in Table 4.

6.3.7 FCS_COP.1/Hash

72 The TOE provides cryptographic hashing services using SHA-1, SHA-256, SHA-384 and SHA-512.

73 SHA is implemented in the following parts of the TSF:

a) SSH (SHA-1, SHA-256, SHA-384, SHA-512);

b) TLS (SHA-1, SHA-256, SHA-384);

c) NTP (SHA384);

d) IPsec (SHA-1, SHA-256, SHA-384, SHA-512);

e) Digital signature verification as part of trusted update validation; and (SHA-256)

f) Hashing of passwords in non-volatile storage. (SHA-512)

74 The relevant NIST CAVP certificate numbers are listed in Table 4.

6.3.8 FCS_COP.1/KeyedHash

75 The TOE provides keyed-hashing message authentication services using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384 and HMAC-SHA-512.

76 HMAC is implemented in SSH, TLS and IPsec.

77 The characteristics of the HMACs used in the TOE are given in Table 19.

Table 19: HMAC Characteristics

Algorithm	Block Size	Key Size	Digest Size
HMAC-SHA-1	512 bits	160 bits	160 bits
HMAC-SHA-256	512 bits	256 bits	256 bits
HMAC-SHA-384	1024 bits	384 bits	384 bits
HMAC-SHA-512	1024 bits	512 bits	512 bits

78 The relevant NIST CAVP certificate numbers are listed in Table 4.

6.3.9 FCS_RBG_EXT.1

79 The TOE contains a CTR_DRBG that is seeded from a CPU provided entropy source. Entropy from the noise is conditioned and used to seed the DRBG with 256 bits of full entropy.

80 Additional detail is provided in the proprietary Entropy Description.

6.3.10 FCS_HTTPS_EXT.1

81 The TOE web GUI is accessed via an HTTPS connection using the TLS implementation described by FCS_TLSS_EXT.1. The TOE's HTTPS protocol complies with RFC 2818.

82 RFC 2818 specifies HTTP over TLS. The majority of RFC 2818 is spent on discussing practices for validating endpoint identities and how connections must be setup and torn down. The TOE web GUI operates on an explicit port designed to natively speak TLS: it does not attempt STARTTLS or similar multi-protocol negotiation which is described in section 2.3 of RFC 2818.

6.3.11 FCS_IPSEC_EXT.1

83 The TOE implements IPsec architecture as specified in RFC 4301. IPsec tunnels are manually set up for EdgeConnect to EdgeConnect VPN traffic, as well as for EdgeConnect to third party VPN traffic. Although several authentication methods are implemented by the TOE, only X509 certificated based authentication has been tested as part of this CC evaluation.

84 Security policy rules for VPN traffic are defined by administrators using the Orchestrator. Each rule specifies a priority value, which establishes the order in which the rules are applied, match criteria and a set of actions (including pass-through, drop or destination IPsec-tunnel). Security policy rules are firewall filtering rules and, as such, they are processed as described in Section 6.5.1. The TOE has a nominal, final rule that matches anything that is otherwise unmatched. During initial configuration of the TOE, the administrators must set the final rule to 'drop'.

85 The TOE implements tunnel mode

86 The TOE supports IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms AES-CBC-128, AES-CBC-256, AES-GCM-128, AES-GCM-256 and Secure Hash algorithms HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512.

87 The TOE implements the IKEv2 protocol defined in RFC 5996 with mandatory NAT support and hash functions defined in RFC 4868.

88 The IKEv2 encrypted payload uses AES-CBC-128, AES-CBC-256, AES-GCM-128, AES-GCM-256 cryptographic algorithms.

89 The IKEv2 SA lifetime is a configurable length of time from 1 minute to 24 hours.

90 The IKEv2 Child SA lifetime is configurable either by number of bytes or length of time between 1 minute and 8 hours.

91 The TOE generates the secret value for IKE Diffie-Hellman key exchange using the RBG specified in FCS_RBG_EXT.1 and has a length of at least 512 bits. The 'x' in g^x mod p is generated using the DRBG in accordance with the negotiated DH group.

92 The TOE generates nonces using the RBG specified in FCS_RGB_EXT.1. Nonces for IKEv2 exchanges have a length according to security strength of the negotiated DH group, and are at least 128 bits in size with at least half the output size of the negotiated PRF hash.

93 The IKE protocol implements the following DH groups:

a) 14 (2048-bit MODP)

b) 15 (3072-bit MODP)

c) 16 (4096-bit MODP)

d) 17 (6144-bit MODP)

e) 18 (8192-bit MODP)

f) 19 (256-bit Random ECP)

g) 20 (384-bit Random ECP)

h) 21 (521-bit Random ECP)

94 The TOE negotiates the DH group configured by the Security Administrator when the IPsec connection is created.

95 The TOE uses CTR_DRBG for the generation of DH exponents and nonces in the IKE key exchange protocol.

96 The length of the exponents is 224 bits (for DH Group 14), 256 bits (for DH Groups 15 and 19), 350 bits (for DH Group 17), 384 bits (for DH Groups 18 and 20), and 512 bits (for DH Group 21).

97 The length of the nonces is 256 bits and meets RFC 5996 requirement of being at least 128 bits and at least half the key size of the negotiated pseudorandom function (PRF).

98 The TOE ensures that the symmetric algorithm strength of the IKEv2 SA connection (112, 128, 192 or 256 bits) is greater or equal to the strength of the symmetric algorithm negotiated for IKEv2 CHILD SA connection.

99 The IKE protocols perform peer authentication using RSA or ECDSA X509v3 certificates according to RFC 4945, with RSA key sizes of 2048 bits, 3072 bits and 4096 bits, and ECDSA key sizes of 256, 384 and 521 bits.

100 When using certificates for peer authentication, the TOE will only establish a trusted channel to peers that provide a valid certificate. The TOE will compare the reference identifier of the peer against the reference identifier stored in the associated certificate. If the two values are not a match, the TOE will not establish the connection. The established trusted channel matches the configured reference identifier with either the DNS or the SAN IP address field type and no other reference identifier type.

6.3.12 FCS_NTP_EXT.1

101 The TOE implements NTP v4 as defined in RFC 5905.

102 The TOE updates its system time SHA384 as the message digest algorithm.

103 The TOE does not update timestamps from NTP broadcast or multicast addresses.

104 The TOE supports configuration of at least 3 NTP time sources.

6.3.13 FCS_SSHS_EXT.1

105 Each TOE component supports the following SSH claims:

106 The TOE supports password-based or public key authentication. In the case of public keys, the TOE authenticates the identity of the SSH client using a local database associating authorized hosts with its corresponding public key.

107 The EdgeConnect supports user public key algorithms rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521.

108 The Orchestrator supports user public key algorithms rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521.

109 The TOE examines the size of each received SSH packet. If the packet is greater than 256 KB, it is automatically dropped.

110 The TOE will re-key SSH connections after 1 hour of after of 1 GB of data has been exchanged (whichever occurs first).

111 The EdgeConnect supports the following SSH claims:

112 Implements SSH in compliance with RFCs 4251, 4252, 4253, 4254, 4256, 4344, 5647, 5656, 8268, 8308 section 3.1 and 8332.

113 Encryption algorithms AES-CTR-128, AES-CTR-256, AES-CBC-128, AES-CBC-256, AES-GCM-128, AES-GCM-256.

114 The supported host key algorithms are rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521.

115 Data integrity MAC algorithms supported are hmac-sha1, hmac-sha2-256, hmac-sha2-512 and implicit.

116 Key Exchange algorithms diffie-hellman-group14-sha1, diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, diffie-hellman-group18-sha512, ecdh-sha2-nistp256, ecdh-sha2-nistp384 and ecdh-sha2-nistp521.

117 The Orchestrator supports the following SSH claims:

118 Implements SSH in compliance with RFCs 4251, 4252, 4253, 4254, 4256, 4344, 5647, 5656, 6668, 8268, 8308 section 3.1 and 8332.

119 Encryptions algorithms AES-CTR-128, AES-CTR-256, AES-GCM-128, AES-GCM-256.

120 The supported host key algorithms for both EdgeConnect and Orchestrator are rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521.

121 Data integrity MAC algorithms HMAC-SHA2-256, HMAC-SHA2-512.

122 Key Exchange algorithms diffie-hellman-group14-sha256, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521.

6.3.14 FCS_TLSC_EXT.1

123 Each TOE component implements TLS 1.2 defined in RFC 5246 and rejects all other TLS and SSL versions.

124 The TLS implementation supports the following ciphersuites:

125 Orchestrator TLS (FTP_ITC.1) and EdgeConnect TLS (FTP_ITC.1):

- TLS_RSA_WITH_AES_128_CBC_SHA

- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_ECDSA_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_RSA_WITH_AES_128_GCM_SHA256
- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

126 EdgeConnect TLS (FPT_ITT.1):

- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
- TLS_DHE_RSA_WITH_AES_256_GCM_SHA384
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_DHE_RSA_WITH_AES_128_GCM_SHA256
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

127 The EdgeConnect (FTP_ITC.1) TLS verifies presented identifiers per RFC 6125 using DNS names in either the CN or SAN or matching an IPv4 address in the SAN. Reference identifiers for Syslog are configured via the Web GUI. The TOE will only support a wildcard in the left-most label (e.g. *.example.com). All other usages of a wildcard will cause a failure in the connection.

128 The Orchestrator (FTP_ITC.1) TLS via the Web GUI verifies presented identifiers per RFC 6125 using DNS names or IPv4 address in either the CN or SAN field. Reference identifiers for Syslog are configured via the Web GUI. The TOE will only support a wildcard in the left-most label (e.g. *.example.com). All other usages of a wildcard will cause a failure in the connection.

129 The Orchestrator (FTP_ITC.1) TLS via the CLI syslog_proxy verifies presented identifiers per RFC 6125 using DNS names in either the CN or SAN field. Reference identifiers for Syslog are configured via the Web GUI. The TOE will only support a wildcard in the left-most label (e.g. *.example.com). All other usages of a wildcard will cause a failure in the connection.

130 The EdgeConnect (FPT_ITT.1) TLS verifies presented identifiers per RFC 6125 using DNS names in the CN or SAN field or matching an IPv4 address in the SAN. Reference identifiers for ITT are configured via the Web GUI. The TOE will only support a wildcard in the left-most label (e.g. *.example.com). All other usages of a wildcard will cause a failure in the connection.

131 The Orchestrator supports the following curves and groups:

a) secp256r1

b) secp384r1

c) secp521r1

d) ffdhe2048

e) ffdhe3072

f) ffdhe4096

132 The EdgeConnect supports the following elliptic curves:

a) secp256r1

b) secp384r1

c) secp521r1

133 Elliptic curves are not configurable on either TOE component.

6.3.15 FCS_TLSS_EXT.1

134 Each TOE component accepts only TLS 1.2 and rejects all other TLS and SSL versions. Specifically, the TLS Server shall reject any client requesting a connection using SSL 2.0, SSL 3.0, TLS 1.0 or TLS 1.1.

135 The EdgeConnect supports the following ciphersuites:

a) TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

b) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

c) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

d) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

e) TLS_DHE_RSA_WITH_AES_256_GCM_SHA384

f) TLS_DHE_RSA_WITH_AES_128_GCM_SHA256

g) TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

h) TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

136 The EdgeConnect supports the following curves and DH parameters:

a) DH parameters of 2048 bits

b) ECDHE curves secp256r1, secp384r1, secp521r1

137 The EdgeConnect supports session resumption using session tickets according to the structural format provided in section 4 of RFC 5077. Session tickets are encrypted using AES-CBC symmetric algorithms, using key size of 128 consistent with FCS_COP.1/DataEncryption. Session resumption is only supported in a single context. When a new session ticket is detected, a full handshake is triggered.

138 The Orchestrator supports the following ciphersuites:

a) TLS_DHE_RSA_WITH_AES_256_GCM_SHA384

b) TLS_DHE_RSA_WITH_AES_128_GCM_SHA256

c) TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

d) TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

e) TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

f) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

g) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

h) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

139 The Orchestrator supports the following curves and DH groups:

a) DH groups ffdhe2048, ffdhe3072, ffdhe4096

b) ECDHE curves secp256r1, secp384r1, secp521r1

140 The Orchestrator does not support session resumption.

6.4 Full Residual Information Protection

6.4.1 FDP_RIP.2

141 The TOE ensures that any previous information content of network packets traversing the TOE is made unavailable upon the deallocation of the memory resources from all associated objects. The TOE uses zeroisation on deallocation of memory resources. The TOE ensures that inactive or terminated network sessions are closed promptly, and all associated resources deallocated.

6.5 Firewall / Packet Filtering

6.5.1 FFW_RUL_EXT.1 / FPF_RUL_EXT.1

142 The TOE permits the configuration of stateful packet filtering policies. The following protocols and associated attributes are configurable within each policy:

a) ICMPv4 (RFC 792)

i) Type; and

ii) Code

b) ICMPv6 (RFC 4443)

i) Type; and

ii) Code

c) IPv4 (RFC 791)

i) Source address;

ii) Destination Address; and

iii) Transport Layer Protocol

d) IPv6 (RFC 2460)

i) Source address;

ii) Destination Address;

iii) Transport Layer Protocol (Next Header)

e) TCP (RFC 793)

i) Source Port; and

ii) Destination Port

f) UDP (RFC 768)

i) Source Port; and

ii) Destination Port

143 Rules can be configured to permit or drop traffic (with the generation of audit log entries for either option).

144 Each rule can be tied to a specific interface (lan1, wan1, etc.).

145 When the TOE boots up, it executes a suite of self-tests. In order for the boot sequence to proceed, each self-test must pass. Network interfaces of the TOE are only activated when all functions required for processing the datagrams are verified and loaded. This ensures that the only when the TOE is fully operational, and all rules enforced before receiving any traffic through the physical interfaces.

146 Each packet that arrives on an interface is subject to the enforcement of stateful traffic filtering. This filtering verifies if the connection is part of an established session or if it is a new connection. If the security attributes of the incoming connection request match those already present for an entry in the state table of the TOE, the information flow is automatically allowed. Otherwise, this is considered a new connection attempt.

147 For a new connection attempt, the packet is compared against the administrator defined rules, and the default if required. Packet rules are enforced in the order defined by the administrator. If no matching rule is found, the TOE will automatically deny the packets and generate a log entry accordingly.

148 The TOE supports the full list of RFC values for IPv4 (RFC 791) and IPv6 (RFC 2460), and is verified by the TOE developer via compliance testing.

149 The session database is consulted to see if an additional session can be created by examining how many currently exist in the database. If this number is below the hardware limit sessions are established by writing the attributes and a TTL into the session database. If the connection is allowed a new session is written into the list of established sessions and can be used to allow subsequent packets for this connection. If logging is enabled for the rule the audit event is sent in real time to the audit server.

150 Any new session will have the first packet of the exchange inspected according to the firewall table as described above, such as the TCP SYN packet during a typical TCP session negotiation for both the sender and receiver. The TOE will write to the session table the expected source and destination ports for this communication flow based on the observed IP headers.

151 The TOE utilizes a session database to track active sessions for TCP, UDP and ICMP (amongst other protocols). The TOE uses source and destination addresses, source and destination ports, sequence number, and individual flags to determine and manage TCP sessions. The TOE uses source and destination addresses and source and destination ports to define and manage UDP flows. The TOE uses source and destination addresses, together with the type and code attributes to manage ICMP active sessions.

152 The TOE will remove existing traffic flows due to session inactivity timeout, or completion of the session.

153 When encountered by the TOE, the following packets will be automatically dropped and an audit log generated for each event:

a) Packets which are invalid fragments (see below);

b) Fragments that cannot be completely re-assembled;

c) Packets where the source address is defined as being on a broadcast network;

d) Packets where the source address is defined as being on a multicast network;

e) Packets where the source address is defined as being a loopback address;

f) Packets where the source or destination address of the network packet is defined as being unspecified (i.e. 0.0.0.0) or an address “reserved for future use” (i.e. 240.0.0.0/4) as specified in RFC 5735 for IPv4;

g) Packets where the source or destination address of the network packet is defined as an “unspecified address” or an address “reserved for future definition and use” (i.e. unicast addresses not in this address range: 2000::/3) as specified in RFC 3513 for IPv6;

h) Packets with the IP options: Loose Source Routing, Strict Source Routing, or Record Route specified.

i) Packets where the source address is equal to the address of the network interface where the network packet was received;

j) Packets where the source or destination address of the network packet is a link local address; and

k) Packets where the source address does not belong to the networks associated with the network interface where the network packet was received.

154 The TOE is capable of detecting fragmented packets. When fragmented packets arrive at their destination, they are reassembled and read. If the fragments do not arrive together, they must be held until all of the fragments arrive. Reassembly of a packet requires all of the fragments. The TOE in the evaluated configuration will attempt to reassemble fragmented packets. When these packets arrive at the TOE they will be held by the TOE for reassembly until the TTL expires. Should the TOE detect that there is a missing or invalid fragment (i.e. first fragment is too small, fragment offset is too small or fragment is out of bounds) during the reassembly the packet will be dropped and logged. IP integrity header checking reads the packets to

verify if a packet is a valid TCP, UDP, and ICMP packets. Verification is also performed to ensure the protocol header is the correct length. This behavior is not capable of being modified or overwritten by the TOE administrator.

155 Incoming packets are inspected against the session database. Sessions that match all the security attributes and do not exceed the TTL are automatically passed on to their destination. Packets that do not match the attributes in the session database are then compared to the defined firewall rules for that interface identifier based on their unique numerical order. Packets that are permitted are passed to their destination, packets marked for logging are written to the audit log and packets marked for dropping are discarded.

156 The TOE maintains half-open TCP sessions in the same manner as full TCP sessions. These TCP sessions are referred to on the TOE as “Embryonic Flows”. The maximum concurrent Embryonic Flows for the EdgeConnect is 256,000. The administrator defines an Embryonic Flow max value as a percentage of the maximum concurrent flows. Once the administrator-defined limit for total sessions is met, sessions (both valid and half-open) are automatically closed based on their timeout value (if not cleared manually by an administrator).

157 All received network packets are processed by the TOE policy engine. The policy engine does stateful filtering of the received network packets according to the configured firewall policies. The TOE kernel monitors the state of any running processes, including the policy engine and VPN processes.

158 The network interfaces of the TOE remain down until the self-tests have passed and all processes are up and running. The failure of any of the self-tests during operation results in the network interfaces being downed and all traffic blocked. During operation, if any of the processes fail or terminate unexpectedly, the kernel will block traffic - i.e. the TOE fails closed.

6.6 Identification and Authentication

6.6.1 FIA_PMG_EXT.1

159 The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”.

160 The minimum password length is settable by the Administrator and can range from 8 to 64 characters.

6.6.2 FIA_UIA_EXT.1

161 Each TOE component requires all users to be successfully identified and authenticated. The TOE warning banner is displayed prior to authentication at each interface.

162 Administrative access to each TOE component is facilitated through several interfaces:

a) **CLI.** Administrative CLI via direct serial connection.

b) **SSH CLI.** Administrative CLI via SSH.

c) **Web GUI.** Administrative interface via HTTPS over TLS.

163 The CLI and Web GUI require the use of usernames and passwords for successful authentication. The SSH CLI requires the use of either usernames and passwords or public key to achieve authentication. Each TOE component maintains an “admin” Security Administrator. The credentials for each interface are identical.

164 For each user enrolled for password-based authentication, the TOE stores a digest of a reference password created when the user selects the password. The entered password is hashed, and the two hashes are compared. If they match, the authentication is considered successful, and the user is granted access to the TOE.

165 For each user enrolled for SSH public key authentication, the TOE stores a reference public key. During the execution of the SSH protocol, the remote user's SSH client sends a digital signature created with the private key of the user. The TOE checks that the signature is valid with respect to the reference public key. If the signature is deemed valid, the authentication is considered successful, and the user is granted access to the TOE.

6.6.3 FIA_UAU_EXT.2

166 Regardless of the interface at which the administrator interacts, the TOE prompts the user for a credential. Only after the administrative user presents the correct authentication credentials will they be granted access to the TOE administrative functionality. No TOE administrative access is permitted until an administrator is successfully identified and authenticated.

167 The TOE provides a local password-based authentication mechanism.

168 The process for authentication is the same for administrative access whether administration is occurring via direct connection or remotely. At initial login, the administrative user is prompted to provide a username. After the user provides the username, the user is prompted to provide the administrative credential associated with the user account (e.g. password). The TOE then either grants administrative access (if the combination of username and credential is correct) or indicates that the login was unsuccessful. The TOE does not provide a reason for failure in the cases of a login failure.

6.6.4 FIA_UAU.7

169 For all authentication at the local CLI the TOE provides no feedback when the administrative password is entered so that the password is obscured.

6.6.5 FIA_AFL.1

170 The TOE is capable of tracking authentication failures of remote administrators.

171 Users identify and authenticate to the TOE using a username and password. The authentication may be locally from a console or remotely from a remote management station. The local console does not implement the lockout mechanism.

172 For each username, the TOE starts a counter for the failed, consecutive authentication attempts. If the authentication attempt fails, the counter value is incremented. If the counter reaches the Administrator-configured maximum value for authentication failures, the offending account is locked for a period of time set by the Administrator. While locked, no authentication attempts are allowed on that account. When an account is locked, other user accounts will remain active, and the locked account shall be unlocked once the locking period expires.

6.6.6 FIA_X509_EXT.1/Rev

173 The TOE performs certificate validation when establishing communication via a trusted channel to a remote syslog server and a VPN peer.

174 The TOE supports certificate validation with a validation path of minimum three certificates, a trusted CA certificate designated as a trust anchor, CA certificates contain basicContraints extensions with a CA flag of TRUE, validation of revocation

status through the use of OCSP as specified in RFC 6960, and validates extendedKeyUsage fields according to the following rules:

a) Server certificates presented for TLS shall have the Server Authentication purpose

b) OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose

175 The TOE obtains OCSP responses from an external HTTP server. The OCSP responder address is read from the leaf and intermediate certificates, and is queried when a connection attempt is made from a TOE component to the external Syslog server or when the EdgeConnect attempts a connection to a VPN peer.

6.6.7 FIA_X509_EXT.1/ITT

176 The TOE supports certificate validation when establishing communication via trusted channel between TOE components.

177 The TOE supports certificate validation with a validation path of minimum two certificates, a trusted CA certificate designated as a trust anchor, CA certificates contain basicContraints extensions with a CA flag of TRUE, revocation using OCSP as specified in RFC 6960 and validates extendedKeyUsage fields according to the following rules:

a) Server certificates presented for TLS shall have the Server Authentication purpose

b) OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose

178 The TOE obtains OCSP responses from an external HTTP server. The OCSP responder address is read from the leaf certificate, and is queried when a connection attempt is made from the EdgeConnect to the Orchestrator component.

6.6.8 FIA_X509_EXT.2

179 The TOE uses X509v3 certificates to support authentication for HTTPS, IPsec and TLS.

180 Certificates are chosen for connection attempts by finding the issuer certificate, specified in the presented leaf or intermediate certificates, in the TOE components trust store.

181 If the validity of a certificate cannot be determined the TOE accepts the certificate.

6.6.9 FIA_X509_EXT.3

182 The Orchestrator is capable of generating Certificate Requests with the following information:

a) Common Name

b) Organization

c) Organizational Unit

d) Country

183 The EdgeConnect is capable of generating Certificate Requests with the following information:

a) Common Name

b) Organization

c) Organizational Unit

184 The TOE only accepts Certificate Responses with the Root CA present as valid.

6.6.10 FIA_PSK_EXT.1 Pre-Shared Key Composition

185 The EdgeConnect component of the TOE accepts generated pre-shared keys used for IPsec and IKEv2.

6.6.11 FIA_PSK_EXT.2 Generated Pre-Shared Keys

186 The pre-shared keys are shared between the EdgeConnect and communicating peers by out of band means.

6.7 Security Management

6.7.1 FMT_MOF.1/ManualUpdate

187 The TOE restricts the ability to perform software updates to Security Administrators.

6.7.2 FMT_MOF.1/Functions

188 The TOE restricts the ability to modify the behaviour of audit functionality when Local Audit Storage Space is full to Security Administrators, for both EdgeConnect and Orchestrator, using the logging configuration options described in the CC Guidance document (see Section 2.4.1).

6.7.3 FMT_MOF.1.1/Services

189 The TOE restricts the ability to start and stop the following services to Security Administrators:

- In the Orchestrator (Via the Web GUI):
  - Syslog: Navigate to **Support => Technical Assistance => Remote Log Receiver.** Click **Edit** on the receiver. Toggle **Enable Receiver** to enable and disable syslog.
- In the EdgeConnect:
  - HTTPS (via the CLI): web https [enable|disable]
  - SSH (via the CLI): ssh server [enable|disable]
  - NTP (via the CLI): ntp [enable|disable]
  - IPSec (via the Web GUI): Navigate to **Administration => System and Networking => Tunnels**. Click **Underlay**, click **Edit** on the tunnel, set **Admin** to Up or Down to enable and disable Underlay tunnels. Click **Passthrough**, click **Edit** on the tunnel, set **Admin** to Up or Down to enable and disable Passthrough tunnels.

6.7.4 FMT_MTD.1/CoreData

190 Users are required to login before being provided with access to any administrative functions. Access to TSF data and functions, including managing the TOE's trust store, is restricted to Security Administrators.

191 The trust store is accessed when administrators import/remove certificates as described in the Common Criteria Guidance document (see Section 2.4.1). By default, only administrators can invoke these functions. The TOE ensures that the trust store is protected through the combination of user authentication and only allowing access to security functions that access trust store data to TOE administrators.

6.7.5 FMT_SMR.2

192 The user account **admin** is a Security Administrator and used to access all interfaces, Web GUI, SSH and local CLI. Each TOE component maintains its own record of the **admin** account.

193 Management of TSF data is restricted to Security Administrators.

6.7.6 FMT_MTD.1/CryptoKeys

194 The TOE restricts the ability to manage SSH keys and X509 Certificates to Security Administrators:

- SSH host key pairs are generated on first boot if it do not exist in manufacturing database. The Security Administrator can request re-generation or import externally generated keys using the CLI or Web GUI as per TOE guidance documentation.
- SSH user public keys for authentication to the SSH server can be similarly generated or imported using the CLI or Web GUI as per TOE guidance documentation.
- TLS authentication key pairs are generated by the TOE automatically. By default, self-signed certificates are created, but certificates can be imported by the Security Administrator using the CLI or Web GUI as per TOE guidance documentation.
- IKE authentication key pairs can be generated or imported by the Security Administrator using the CLI or Web GUI as per TOE guidance documentation.

195 Asymmetric key pair generation uses FIPS Approved SP800-90A DRBG in compliance with FIPS 186-4 RSA or ECDSA key pair generation methods. The TOE implements the creation Certificate Signing Requests (CSRs) to support X509 certificate generation.

6.7.7 FMT_SMF.1 / FMT_SMF.1/VPN / FMT_SMF.1/FFW

196 The TOE may be managed via the CLI (console & SSH) or GUI (HTTPS). The specific management capabilities include:

Table 20: TOE Component Management Capabilities

Management Capability	TOE Components	Orchestrator Interfaces	EdgeConnect Interfaces
Ability to administer the TOE locally and remotely	All	CLI and GUI	CLI and GUI
Ability to configure the access banner (FTA_TAB.1)	All	CLI and GUI	CLI and GUI

Management Capability	TOE Components	Orchestrator Interfaces	EdgeConnect Interfaces
Ability to configure the session inactivity time before session termination or locking (FTA_SSL_EXT.1, FTA_SSL.3)	All	CLI and GUI	CLI and GUI
Ability to update the TOE and to verify the updates (FMT_MTD.1/ManualUpdate, FPT_TUD_EXT.1)	All	CLI	CLI and GUI
Ability to configure the authentication failure parameters (FIA_AFL.1)	All	CLI and GUI	CLI and GUI
Ability to start and stop services	All	CLI	CLI
Ability to configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full)	EdgeConnect	N/A	CLI and GUI
Ability to manage the cryptographic keys (FMT_MTD.1/CryptoKeys, FCS_CMK.1)	All	CLI	CLI
Ability to configure the cryptographic functionality (FCO_CPC_EXT.1)	EdgeConnect	N/A	GUI
Ability to set the time which is used for time-stamps	All	CLI	CLI
Ability to configure NTP	All	CLI	CLI
Ability to configure the lifetime for IPsec SAs	All	GUI	GUI
Ability to configure the reference identifier for the peer	All	GUI	GUI
Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors	All	CLI	CLI
Ability to import X.509v3 certificates to the TOE's trust store	All	CLI	CLI
Ability to manage the trusted public keys database	All	CLI	CLI
Ability to configure firewall rules (per FMT_SMF.1/FFW)	All	GUI	GUI
Definition of packet filtering rules (per FMT_SMF.1/VPN)	All	GUI	GUI

Management Capability	TOE Components	Orchestrator Interfaces	EdgeConnect Interfaces
Association of packet filtering rules to network interfaces (per FMT_SMF.1/VPN)	All	GUI	GUI
Ordering of packet filtering rules by priority (per FMT_SMF.1/VPN)	All	GUI	GUI

197 Configuration activities for FMT_SMF.1/VPN and FMT_SMF.1/FFW can be performed by the Orchestrator for the purposes of communicating to and enforcement by the EdgeConnect.

6.8 Protection of the TSF

6.8.1 FPT_ITT.1

198 The TOE provides protection from disclosure of internal TOE data transfers through the use of TLS.

6.8.2 FPT_SKP_EXT.1

199 Keys are protected as described in Table 21. In all cases, plaintext keys cannot be viewed through an interface designed specifically for that purpose.

Table 21: Keys

Key	Algorithm	Storage	Zeroization
SSH Private Host Keys	RSA / ECDSA	Flash – plaintext	Keys are destroyed when generating new keys by deleting the previous file and creating a new file. Initiated via CLI command by the Security Administrator.
SSH User Public Keys	RSA / ECDSA	Flash – plaintext	Zeroize upon FIPS secure erase operation.
SSH Ephemeral Keys	AES / DH / ECDH	RAM – plaintext	The cryptographic module ensures that keys (including re-keyed keys) are overwritten with zeroes upon termination of SSH session or TOE component restart .
TLS Private Keys	RSA / ECDSA	Flash - plaintext	Keys are deleted and zeroized from the trust store when deletion is initiated by the Security Administrator via the Web GUI.
TLS Public Keys	RSA / ECDSA	Flash - plaintext	Zeroize in RAM upon termination of TLS session or module restart. Zeroize in hard drive upon invocation of FIPS secure erase operation.
TLS Ephemeral Keys	AES / ECDH	RAM – plaintext	The cryptographic module ensures that keys are overwritten with zeroes upon

Key	Algorithm	Storage	Zeroization
			termination of TLS session or TOE component restart.
IPsec Ephemeral Keys	AES / DH	RAM – plaintext	Zeroize upon deletion of IPSec tunnel or module restart.
IKE Private Keys	RSA / ECDSA	Flash - plaintext	Keys are deleted and zeroized from the trust store when deletion is initiated by the Security Administrator via the Web GUI.
IKE Public Keys	RSA / ECDSA	Flash - plaintext	Zeroize upon deletion of IPSec tunnel or module restart.
DRBG Internal state	V and key are used as part of HMAC and CTR DRBG process. V and C are used as part of HASH DRBG process.	Plaintext in RAM	Zeroize by calling fips/rand/fips_drbg_lib.c:FIPS_ drbg_free() or by rebooting the module.
DRBG Entropy	Entropy input strings used as part of the DRBG process.	Plaintext in RAM	Zeroize by rebooting the module.
Firmware verification key	ECDSA	Hardcoded	Zeroize upon FIPS secure erase operation.

6.8.3 FPT_APW_EXT.1

200 Passwords are protected as describe in Table 22. In all cases plaintext passwords cannot be viewed through an interface designed specifically for that purpose.

Table 22: Passwords

Key/Password	Generation/ Algorithm	Storage
Locally stored administrator passwords	User generated	Flash - SHA-512 hash

6.8.4 FPT_FLS.1/SelfTest

201 The cryptographic functionality when the tests fail, the boot operation will fail and not complete. The power on integrity checks of the TOE fail, the boot up operation will not complete. When the noise source tests fail, the boot operation will fail and not complete. When the health test of the noise source fails, the TOE's boot operation will fail and not complete. The TOE does not complete the boot process in any

instance when the tests fail and therefore does not require enforcing security policies outside of successful tests and completed boot operations. When the device completes the boot up operation, this is evidence that the self-tests have passed, and that the TOE, and the cryptographic functions are operating correctly.

6.8.5 FPT_TST_EXT.1

202 Each TOE component performs the following tests individually at start-up:

a) Memory Basic Input/Output System (BIOS) self-tests by performing a series of writing and reading data to and from memory.

b) Boot loader image verification – the boot loader performs a digital signature verification check of the image of the TOE prior to booting.

c) Cryptographic known answer tests and integrity tests are performed on the following cryptographic modules as per FIPS 140-2 requirements:

   i) Silver Peak EdgeConnect Cryptographic library, Crypto Library 2021 version 1.1

   ii) HPE BC-FJA (Bouncy Castle FIPS Java API), version 1.0.2

   iii) HPE Aruba Networking Orchestrator Cryptographic Library, Crypto Library 2024 version 1.0

203 If any of the tests fails, the TOE aborts the booting of the device.

6.8.6 FPT_TST_EXT.3

204 When loaded for execution, each TOE component runs a suite of integrity verification self-tests through the cryptographic service specified in FCS_COP.1/SigGen to demonstrate the correct operation of the system.

205 Self-tests are run by performing a digital signature verification check of the TOE firmware using a 4096 bit RSA signature.

6.8.7 FPT_TUD_EXT.1

206 The most recently installed version of the Orchestrator is displayed in upper right corner of the UI browser window. The most recently installed version of the EdgeConnect may be queried using any administrative interface. Updates to the EdgeConnect may be executed with delayed activation and the previous TOE version may be queried using “show image” on the CLI, or on the Web GUI via Administration/SOFTWARE/Upgrade/Software Versions/.

207 The Orchestrator supports automatically checking for updates for both Orchestrator and EdgeConnect software updates.

208 The EdgeConnect supports manually initiating updates and no other mechanism.

209 The Security Administrator manually initiates updates on the Orchestrator from the CLI. TOE update files must first be copied to the TOE via SCP.

210 The Security Administrator manually initiates updates on the EdgeConnect from the Web GUI. The update files are copied to the TOE via HTTPS/TLS

211 TOE update files are digitally signed (ECDSA) and the signature is verified using a hardcoded public key prior to installation of the update. If verification fails, the update is aborted, and an error message is displayed. If verification succeeds for the Orchestrator, the component is rebooted and the new image becomes active. If verification succeeds for the EdgeConnect, the component will either automatically reboot and apply the new image or install the new image but remain operational on

the previous image until a Security Administrator specifies the new image as the active partition and manually reboots.

6.8.8 FPT_STM_EXT.1

212 The TOE incorporates an internal clock for each TOE component which is free from outside interference. The hardware models have an internal battery-backed hardware clock for reliability. The EdgeConnect makes use of manual time setting and synchronizing time with an NTP server to maintain date and time. The Orchestrator makes use of synchronizing time with an NTP server to maintain date and time. The external NTP servers are assumed to be reliable authorities of time.

213 The TOE makes use of time for the following:

a) Audit record timestamps

b) Session timeouts (lockout enforcement)

c) Determining X.509 expiration validation

d) Rekeying for SSH and IKE/IPsec connections

6.9 TOE Access

6.9.1 FTA_SSL_EXT.1

214 Each TOE component supports session termination of interactive local sessions. The Security Administrator may configure the TOE to terminate an inactive local interactive session following a Security Administrator specified period of inactivity, defined in minutes. Each TOE component stores its own inactivity time period configuration. This is applicable to the local CLI. Idle timeout for the local CLI can be configured between 1 to 120 minutes.

6.9.2 FTA_SSL.3

215 Each TOE component supports session termination of interactive remote sessions. The Security Administrator may configure the TOE to terminate an inactive remote interactive session following a Security Administrator specified period of inactivity, defined in minutes. Each TOE component stores its own inactivity time period configuration. This is applicable to the SSH CLI and Web GUI. Idle timeout for the SSH CLI can be configured between 1 to 120 minutes. Idle timeout for the Web GUI can be configured between 1 to 60 minutes.

6.9.3 FTA_SSL.4

216 Administrative users may terminate their own sessions at any time by either calling the “exit” command at the local CLI and remote SSH CLI or by using the “Logout” button at the Web GUI.

6.9.4 FTA_TAB.1

217 The TOE displays an administrator configurable message to users prior to login at the CLI, SSH CLI, and Web GUI. The banner gets set in a template which is applied to Orchestrator and managed EdgeConnect appliances and is the same for all methods of accessing the TOE.

6.10 Trusted Path/Channels

6.10.1 FTP_ITC.1

218 The TOE supports secure communication using TLS between itself and an external audit server per FCS_TLSC_EXT.1. When the TOE and the audit server establish an TLS connection, the TOE initiates the connection.

6.10.2 FTP_ITC.1/VPN

219 The TOE is capable of using IPsec in tunnel mode to provide a communication channel between itself and IPsec peers per FCS_IPSEC_EXT.1. The TOE permits IPsec peers to initiate communication as well as initiating communication to the IPsec peers.

6.10.3 FTP_TRP.1/Admin

220 The TOE provides the following trusted paths for remote administration:

a) **SSH CLI.** Administrative CLI via SSH per FCS_SSHS_EXT.1.

b) **Web GUI.** HTTPS GUI via HTTPS using TLS per FCS_HTTPS_EXT.1.

7 Rationale

7.1 Conformance Claim Rationale

221 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 distributed network device, consistent with Use Case 3 of CPP_ND_V2.2E. The TOE acts as a security gateway implementing both stateful traffic filtering and multi-site VPN functionality, consistent with MOD_CPP_FW_V1.4e and MOD_VPNGW_V1.3, respectively.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are reproduced directly from the CPP_ND_V2.2E, MOD_CPP_FW_V1.4e and MOD_VPNGW_V1.3.

c) **Security objectives.** As shown in section 4, the security objectives are reproduced directly from the CPP_ND_V2.2E, MOD_CPP_FW_V1.4e and MOD_VPNGW_V1.3.

d) **Security requirements.** As shown in section 5, the security requirements are reproduced directly from the CPP_ND_V2.2E, MOD_CPP_FW_V1.4e and MOD_VPNGW_V1.3. No additional requirements have been specified.

7.2 Security Objectives Rationale

222 All security objectives are drawn directly from the CPP_ND_V2.2E, MOD_CPP_FW_V1.4e and MOD_VPNGW_V1.3.

7.3 Security Requirements Rationale

223 All security requirements are drawn directly from the CPP_ND_V2.2E, MOD_CPP_FW_V1.4e and MOD_VPNGW_V1.3. Table 23 presents a mapping between threats and SFRs as presented in the CPP_ND_V2.2E, MOD_CPP_FW_V1.4e and MOD_VPNGW_V1.3.

Table 23: CPP_ND_V2.2E SFR Rationale

Identifier	SFR Rationale
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	• The Administrator role is defined in FMT_SMR.2 and the relevant administration capabilities are defined in FMT_SMF.1 and FMT_MTD.1/CoreData, with optional additional capabilities in FMT_MOF.1/Services and FMT_MOF.1/Functions • The actions allowed before authentication of an Administrator are constrained by FIA_UIA_EXT.1, and include the advisory notice and consent warning message displayed according to FTA_TAB.1 • The requirement for the Administrator authentication process is described in FIA_UAU_EXT.2 • Locking of Administrator sessions is ensured by FTA_SSL_EXT.1 (for local sessions), FTA_SSL.3 (for

Identifier	SFR Rationale
	remote sessions), and FTA_SSL.4 (for all interactive sessions) • The secure channel used for remote Administrator connections is specified in FTP_TRP.1/Admin • (Malicious actions carried out from an Administrator session are separately addressed by T.UNDETECTED_ACTIVITY) • (Protection of the Administrator credentials is separately addressed by T.PASSWORD_CRACKING).
T.WEAK_CRYPTOGRAPHY	• Requirements for key generation and key distribution are set in FCS_CKM.1 and FCS_CKM.2 respectively • Requirements for use of cryptographic schemes are set in FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, and FCS_COP.1/KeyedHash • Requirements for random bit generation to support key generation and secure protocols (see SFRs resulting from T.UNTRUSTED_COMMUNICATION_CHANNELS) are set in FCS_RBG_EXT.1 • Management of cryptographic functions is specified in FMT_SMF.1
T.UNTRUSTED_COMMUNICATION_CHANNELS	• The general use of secure protocols for identified communication channels is described at the top level in FTP_ITC.1 and FTP_TRP.1/Admin; for distributed TOEs the requirements for inter-component communications are addressed by the requirements in FPT_ITT.1 • Requirements for the use of secure communication protocols are set for all the allowed protocols in FCS_HTTPS_EXT.1, FCS_IPSEC_EXT.1, FCS_SSHS_EXT.1, FCS_TLSC_EXT.1, FCS_TLSS_EXT.1 • Optional and selection-based requirements for use of public key certificates to support secure protocols are defined in FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3
T.WEAK_AUTHENTICATION_ENDPOINTS	• The use of appropriate secure protocols to provide authentication of endpoints (as in the SFRs addressing T.UNTRUSTED_COMMUNICATION_CHANNELS) are ensured by the requirements in FTP_ITC.1 and FTP_TRP.1/Admin; for distributed TOEs the authentication requirements for endpoints in inter-component communications are addressed by the requirements in FPT_ITT.1 • Additional possible special cases of secure authentication during registration of distributed TOE components are addressed by FCO_CPC_EXT.1.

Identifier	SFR Rationale
T.UPDATE_COMPROMISE	• Requirements for protection of updates are set in FPT_TUD_EXT.1 • Requirements for management of updates are defined in FMT_SMF.1 and (for manual updates) in FMT_MOF.1/ManualUpdate, with optional requirements for automatic updates in FMT_MOF.1/AutoUpdate
T.UNDETECTED_ACTIVITY	• Requirements for basic auditing capabilities are specified in FAU_GEN.1 and FAU_GEN.2, with timestamps provided according to FPT_STM_EXT.1 and if applicable, protection of NTP channels in FCS_NTP_EXT.1 • Requirements for protecting audit records stored on the TOE are specified in FAU_STG.1 • Requirements for secure transmission of local audit records to an external IT entity via a secure channel are specified in FAU_STG_EXT.1 • If (optionally) configuration of the audit functionality is provided by the TOE then this is specified in FMT_SMF.1, and confining this functionality to Security Administrators is required by FMT_MOF.1/Functions.
T.SECURITY_FUNCTIONALITY_COMPROMISE	• Protection of secret/private keys against compromise is specified in FPT_SKP_EXT.1 • Secure destruction of keys is specified in FCS_CKM.4 • If (optionally) management of keys is provided by the TOE then this is specified in FMT_SMF.1, and confining this functionality to Security Administrators is required by FMT_MTD.1/CryptoKeys • (Protection of passwords is separately covered under T.PASSWORD_CRACKING)
T.PASSWORD_CRACKING	• Requirements for password lengths and available characters are set in FIA_PMG_EXT.1 • Protection of password entry by providing only obscured feedback is specified in FIA_UAU.7 • Actions on reaching a threshold number of consecutive password failures are specified in FIA_AFL.1 • Requirements for secure storage of passwords are set in FPT_APW_EXT.1.
T.SECURITY_FUNCTIONALITY_FAILURE	• Requirements for running self-test(s) are defined in FPT_TST_EXT.1
P.ACCESS_BANNER	• An advisory notice and consent warning message is required to be displayed by FTA_TAB.1

Identifier	SFR Rationale
T.DATA_INTEGRITY	• The threat of data integrity compromise is a specific example of the T.WEAK_CRYPTOGRAPHY threat defined in the Base-PP.
T.NETWORK_ACCESS (VPNGW)	• The threat of a malicious entity accessing protected network resources without authorization is a specific example of the T.UNTRUSTED_COMMUNICATION_CHANNELS threat defined in the Base-PP.
T.NETWORK_ACCESS (FFW)	• Requirements to prevent unauthorised access to protected devices and services are defined in FFW_RUL_EXT.1 and supported by FMT_SMF.1/FFW
T.NETWORK_DISCLOSURE (VPNGW)	• Exposure of network devices due to insufficient protection is a specific example of the T.UNTRUSTED_COMMUNICATION_CHANNELS threat defined in the Base-PP.
T.NETWORK_DISCLOSURE (FFW)	• Requirements to prevent unauthorised disclosure of network information are defined in FFW_RUL_EXT.1 and supported by FMT_SMF.1/FFW.
T.NETWORK_MISUSE (VPNGW)	• Depending on the specific nature of the misuse of network resources, this threat is a specific manifestation of either the T.UNTRUSTED_COMMUNICATION_CHANNELS or T.WEAK_AUTHENTICATION_ENDPOINTS threat defined in the Base-PP.
T.NETWORK_MISUSE (FFW)	• Requirements to prevent network misuse traffic are defined in FFW_RUL_EXT.1 and supported by FMT_SMF.1/FFW • Requirements to prevent the unintended dissemination of data from packets after deletion are defined in FDP_RIP.2
T.REPLAY_ATTACK	• A replay attack is mentioned in the Base-PP as a specific type of attack based on the T.UNTRUSTED_COMMUNICATION_CHANNELS threat.
T.MALICIOUS_TRAFFIC	• Requirements to prevent malformed traffic are defined in FFW_RUL_EXT.1