Gigamon GigaVUE Fabric Manager v6.6

Security Target

ST Version: 1.0
December 28, 2024

Prepared For:

Gigamon Inc.
3300 Olcott Street
Santa Clara, CA 95054

Prepared By:

Booz | Allen | Hamilton
delivering results that endure

Cyber Assurance Testing Laboratory
1100 West St
Laurel, MD 20707

Table of Contents

List of Figures

List of Tables

1 Security Target Introduction

This chapter presents the Security Target (ST) identification information and an overview. An ST contains the Information Technology (IT) security requirements of an identified Target of Evaluation (TOE) and specifies the functional and assurance security measures offered by the TOE.

1.1 ST Reference

This section provides information needed to identify and control this ST and its Target of Evaluation.

1.1.1 ST Identification

ST Title: Gigamon GigaVUE Fabric Manager v6.6 Security Target
ST Version: 1.0
ST Publication Date: December 28, 2024
ST Author: Booz Allen Hamilton

1.1.2 Document Organization

Chapter 1 of this document provides identifying information for the ST and TOE as well as a brief description of the TOE and its associated TOE type.

Chapter 2 describes the TOE in terms of its physical boundary, logical boundary, exclusions, and dependent Operational Environment components.

Chapter 3 describes the conformance claims made by this ST.

Chapter 4 describes the threats, assumptions, objectives, and organizational security policies that apply to the TOE.

Chapter 5 defines extended Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs).

Chapter 6 describes the SFRs that are to be implemented by the TSF.

Chapter 7 describes the SARs that will be used to evaluate the TOE.

Chapter 8 provides the TOE Summary Specification, which describes how the SFRs that are defined for the TOE are implemented by the TSF.

1.1.3 Terminology

This section defines the terminology used throughout this ST. The terminology used throughout this ST is defined in Table 1-1 and 1-2. These tables are to be used by the reader as a quick reference guide for terminology definitions.

Term	Definition
Admin	A user who is assigned the “Admin” role on the TOE’s CLI and has the ability to manage the TSF. Synonymous with Security Administrator.
Local CLI	Synonymous with the term “local console”.
Super Admin	A user who is assigned the “Super Admin” role on the TOE’s Web GUI and has the ability to manage the TSF. Synonymous with Security Administrator.

Table 1-1: Customer Specific Terminology

Term	Definition
Credential	Data that establishes the identity of a user (e.g., a cryptographic key or password).
Operating System (OS)	Software that manages hardware resources and provides services for applications.
Platform	A platform can be an operating system, hardware environment, a software-based execution environment, or some combination of these. These types of platforms may also run atop other platforms.
Security Administrator	An authorized administrator role that is authorized to manage the TOE and its data. This TOE defines three separate user roles, but only the most privileged role (Admin) is authorized to manage the TOE's security functionality and is therefore considered to be the Security Administrator for the TOE.
Trusted Channel	An encrypted connection between the TOE and a system in the Operational Environment.
Trusted Path	An encrypted connection between the TOE and the application a Security Administrator uses to manage it (SSH client, terminal client, etc.).
User	In a CC context, any individual who has the ability to access the TOE functions or data.

Table 1-2: CC Specific Terminology

1.1.4 Acronyms

The acronyms used throughout this ST are defined in Table 1-3. This table is to be used by the reader as a quick reference guide for acronym definitions.

Acronym	Definition
AES	Advanced Encryption Standard
API	Application Programming Interface
CA	Certificate Authority
CAVP	Cryptographic Algorithm Verification Program
CBC	Cipher Block Chaining
CC	Common Criteria
CLI	Command-Line Interface
cPP	collaborative Protection Profile
CPU	Central Processing Unit
CRL	Certificate Revocation List
CSP	Cryptographic Service ProviderIDS
CTR	Counter
DRBG	Deterministic Random Bit Generator
FM	Fabric Manager
FTP	File Transfer Protocol
GMC	Galois/Counter Mode
HMAC	Hash-based Message Authentication Code
HTTPS	Hypertext Transfer Protocol Secure
I&A	Identity and Access
IDS	Intrusion Detection System
MAC	Message Authentication Code
NIAP	National Information Assurance Partnership
NTP	Network Time Protocol

OCSP	Online Certificate Status Protocol
OS	Operating System
PP	Protection Profile
RAM	Random Access Memory
RBG	Random Bit Generator
RNG	Random Number Generator
RU	Rack Unit
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSH	Secure Shell
ST	Security Target
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Function
UI	User Interface

Table 1-3: Acronym Definition

1.1.5 References

[1] collaborative Protection Profile for Network Devices Version 2.2e 20200327 [NDcPP]

[2] Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated April 2017, version 3.1, Revision 5, CCMB-2017-004-001

[3] Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated April 2017, version 3.1, Revision 5, CCMB-2017-004-002

[4] Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated April 2017, version 3.1, Revision 5, CCMB-2017-004-003

[5] Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated April 2017, version 3.1, Revision 5, CCMB-2017-004-004

1.2 TOE Reference

The TOE is the Gigamon GigaVUE Fabric Manager v6.6. The TOE may also be referred to as Gigamon-FM throughout this document.

1.3 TOE Overview

The Gigamon-FM is a network device that includes both hardware and software and is used to manage other Gigamon hardware devices. Gigamon-FM's primary functionality is to offer a central location for the configuration, management, and operation of the Gigamon Deep Observability Pipeline which provides network visibility across physical, virtual, and cloud infrastructure. Gigamon-FM allows for the configuring traffic policies, visualizing network topology connectivity, and identifying visibility hot spots within a network.

The following figure depicts the TOE boundary.

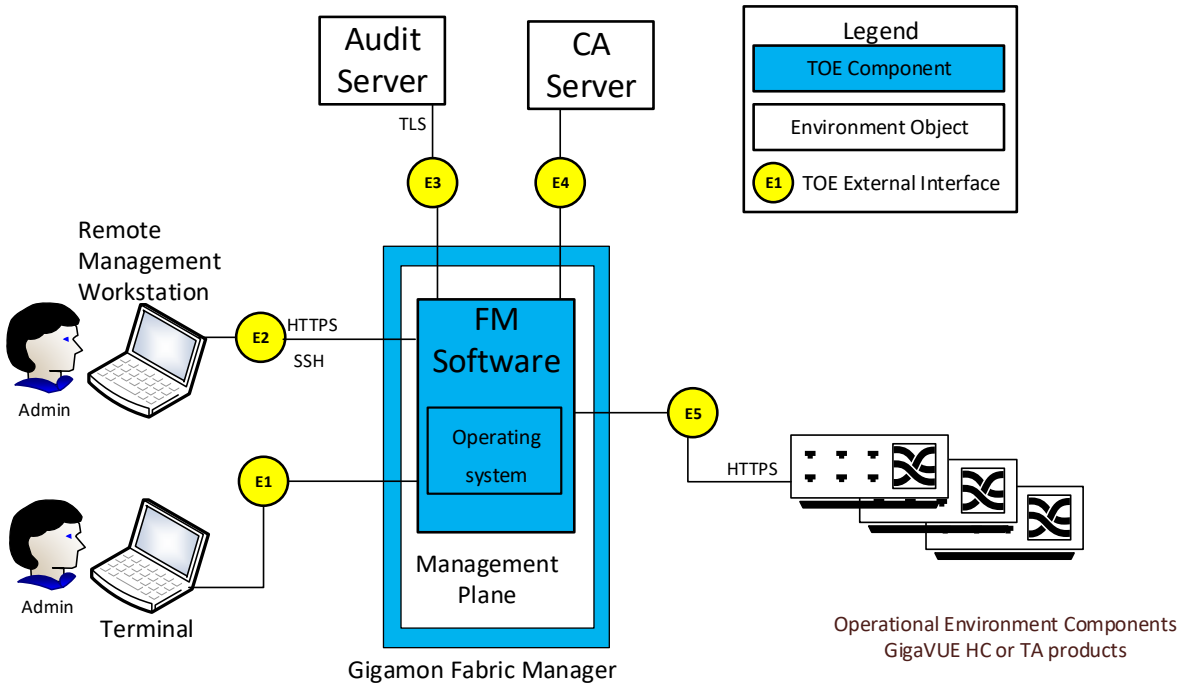

Figure 1: TOE Boundary for Gigamon-FM

As illustrated in Figure 1, the TOE is a single hardware device that has management ports, network (or ingress) ports, and tool (or egress) ports.

The external interfaces that are relevant to the TOE boundary are depicted as an E# in a yellow circle in the figure above. These interfaces are established via a dedicated Management Ethernet Port and collectively is referred to as the management plane. The relevant external interfaces are:

- E1 – This is the local Security Administrator access to the CLI via a direct connection.
- E2 – The TOE acts as a SSH server for remote Security Administrator access to the CLI.
- E2 – The TOE acts as an HTTPS/TLS server for remote Security Administrator access to the Web GUI.
- E3 – The TOE acts as an TLS client for sending audit records to a remote audit server for external audit log storage.
- E4 – The TOE interfaces with a Certification Authority (CA) for issuance of server certificates and publication of a Certificate Revocation List (CRL) to determine the validity of certificates presented to the TOE.
- E5 – The TOE acts as a HTTPS/TLS Client for trusted communication to GigaVUE appliances (Operational Environment Component). Gigamon-FM is only compatible with the Gigamon GigaVUE HA series and TA series appliances.

1.4 TOE Type

The TOE type for this product is Network Device. The product is a hardware appliance whose primary functionality is related to the handling of network traffic.

The NDcPP defines a network device as "a device that is connected to a network and has an infrastructure role within that network…Under this cPP, NDs may be physical or virtualized. A physical Network Device (pND) consists of network device functionality implemented inside a physical chassis with physical network connections. The network device functionality may be implemented in either hardware or software or both. For pNDs, the TOE encompasses the entire device—including both the network device functionality and the physical chassis. There is no distinction between TOE and TOE Platform."

The TOE is a standalone network device composed of hardware and software. It is connected to an enterprise's network to configure traffic policies, visualize network topology connectivity, and identify visibility hot spots within a network. Gigamon-FM plays an infrastructure role in an enterprise network through its ability to configure and enforce traffic flow policies as well as to provide visibility into the enterprise's network.

2 TOE Description

This section provides a description of the TOE in its evaluated configuration. This includes the physical and logical boundaries of the TOE.

2.1 Evaluated Components of the TOE

The following table describes the TOE components. There is only one model in the evaluated configuration:

Component	Definition
Gigamon-FM	1RU appliance running Gigamon-FM software v6.6

Table 2-1: Evaluated Components of the TOE

2.2 Components and Applications in the Operational Environment

The following table lists components and applications in the TOE's operational environment that must be present for the TOE to be operating in its evaluated configuration:

Component	Definition
Certification Authority (CA)	A server that acts as a trusted issuer of digital certificates and distributes a CRL that identifies revoked certificates. Represented by E5 in the figure above.
Management Workstation	Any general-purpose computer that is used by a Security Administrator to manage the TOE. The TOE can be managed remotely, in which case the management workstation requires an SSH client to access the CLI or a web browser to access the Web GUI. The TOE can also be managed locally, in which case the management workstation must be physically connected to the TOE using the serial port and must use a terminal emulator that is compatible with serial communications. Represented by E1, E2, & E3 in the figure above.

Audit Server	The audit server connects to the TOE and allows the TOE to send syslog messages to it for remote storage. This is used to send copies of audit data to be stored in a remote location for data redundancy purposes. Represented by E4 in figure above.
Gigamon GigaVUE Appliances	The Gigamon GigaVUE appliances are separately evaluated products (VID11487) that Gigamon-FM can manage over a secure channel. Represented by E6 in figure above.

Table 2-2: Components of the Operational Environment

2.3 Excluded from the TOE

The following optional products, components, and/or applications can be integrated with the TOE but are not included in the evaluated configuration. They provide no added security related functionality for the evaluated product. They are separated into three categories: not installed, installed but requires a separate license, and installed but not part of the TSF.

2.3.1 Not Installed

There are no optional components that are omitted from the installation process.

2.3.2 Installed but Requires a Separate License

There are no excluded components that are installed and require a separate license.

2.3.3 Installed but Not Part of the TSF

The TOE includes a number of functions that are outside the scope of the claimed Protection Profile. These functions cannot be mapped to any NDcPP SFRs. The functions include the ability to manage and configure, monitor, and control HC1 and TA Gigamon GigaVUE appliances across a physical, virtual, and cloud environment, providing a unified view of network traffic and enabling policy management for security and network operations teams by directing traffic to the appropriate monitoring tools.

The only TOE functions evaluated are those claimed and described in this ST.

2.4 Physical Boundary

2.4.1 Hardware

The Gigamon-FM specific hardware and configurations are as follows:

Property	Gigamon-FM
Model/Part Number	GFM-HW1-FM010
Size	One rack unit (1RU)
Processor	Dual Intel Xeon Silver 4114 2.1GHz, 8C/16T
Management	IPMI 2.0 compliant 2 x 1/10G SFP+ 2 x 100/1000M Base-T LAN Serial console (115,200 baud)

Property	Gigamon-FM
Connectors	Back: 4 x 10/100/1000Mbps LOM 1 x 10/100/1000Mbps iDRAC9 Enterprise 1 x DB9 serial 1 x USB 3.0, one USB 2.0 1 x DB15 VGA Front: 2 x USB 2.0 (disabled in BIOS) 1 x DB15 VGA

Table 2-3: Gigamon-FM Properties

2.4.2 Software

Gigamon-FM runs Gigamon software v6.6 which includes the Rocky Linux 8.10 operating system.

2.5 Logical Boundary

The TOE is comprised of several security features. Each of the security features identified above consists of several security functionalities, as identified below.

- Security Audit
- Cryptographic Support
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels

2.5.1 Security Audit

Audit records are generated for various types of management activities and events. The audit records include the date and time stamp of the event, the event type and subject identity. In the evaluated configuration, the TSF is configured to transmit audit data to a remote audit server using TLS. Audit data is also stored locally to ensure availability of the data if communications with the audit server become unavailable.

2.5.2 Cryptographic Support

The TOE uses sufficient security measures to protect its data in transmission by implementing cryptographic methods and trusted channels. The TOE uses:

- SSH to secure the remote CLI.
- HTTPS to secure the connection to the Web GUI and to the GigaVUE appliances.
- TLS to secure the connection to the audit server.

Cryptographic keys are generated using the Hash_DRBG provided by this module. The TOE destroys plaintext and private keys in both volatile and non-volatile storage.

The following table contains the CAVP algorithm certificates:

SFR	Algorithm	CAVP Cert. #
FCS_CKM.1 - ECC key generation schemes	ECDSA KeyGen (FIPS186-4) P-256, P-384, and P-521 ECDSA KeyVer (FIPS186-4) P-256, P-384, and P-521	A6377
FCS_CKM.2 – ECDSA key establishment	KAS-ECC-SSC Sp800-56Ar3	A6377
FCS_COP.1/DataEncryption	AES CBC 128 bits and 256 bits AES CTR 128 bits and 256 bits AES GCM 128 bits and 256 bits	A6377
FCS_COP.1/Hash	SHA-256, SHA-384, and SHA-512	A6377
FCS_COP.1/KeyedHash	HMAC-256, HMAC-384, and HMAC-512	A6377
FCS_COP.1/SigGen - ECDSA	ECDSA SigGen (FIPS186-4) P-256, P-384, and P-521 ECDSA SigVer (FIPS186-4) P-256, P-384, and P-521	A6377
FCS_RBG_EXT.1	Hash DRBG	A6377

Table 2-4: Cryptographic Algorithm Table

2.5.3 Identification and Authentication

All users must be identified and authenticated to the TOE before being allowed to perform any actions on the TOE. This is true of users accessing the TOE via the local CLI, the Web GUI via HTTPS, or the protected path using the remote CLI via SSH. Users authenticate to the TOE using one of the following methods:

- Username/password (Web GUI and SSH)
- Username/public key (SSH only)

The TSF provides a configurable number of maximum consecutive authentication failures that are permitted by a user. Once this number has been met, the account is locked until a manual unlock occurs for Web GUI users or an administratively set time for CLI users. Passwords that are maintained by the TSF can be composed of upper case, lower case, numbers and special characters. The Security Administrator can define the minimum password length between 8 and 64 characters. Password information is never revealed during the authentication process including during login failures. Before a user authenticates to the device, a configurable warning banner is displayed.

As part of establishing trusted remote communications, the TOE provides X.509 certificate functionality. In addition to verifying the validity of certificates, the TSF can check their revocation status using a certificate revocation list (CRL).

2.5.4 Security Management

The TOE has two roles to fulfill the role of Security Administrator: Admin and Super Admin. The Admin is the administrative role for the local CLI and remote CLI. The Super Admin is the administrative role for the Web GUI. Management functions can be performed using the local CLI, remote CLI, and Web GUI. Both Security Administrative roles is able to perform all security-relevant management functionality (such as user management, password policy configuration, application of software updates, and configuration of cryptographic settings). All software updates to the TOE can only be performed manually by an Admin role user.

2.5.5 Protection of the TSF

The TOE stores the hashed representation of passwords using SHA-512. Keys are stored in an encrypted internal database that is integrity checked at boot time. The TOE has a hardware clock that is used for keeping time. The time can be manually set by the Security Administrator. The TOE executes a suite of self-tests during boot and at the request of an Security Administrator.

The version of the TOE (both the currently executing version and the latest installed/updated version) can be obtained by an Admin role user from the CLI interface. The updated image is verified through manually validating the correct published hash.

2.5.6 TOE Access

The TOE can terminate inactive local CLI, remote CLI, or Web GUI sessions after a specified time period. Users can also terminate their own interactive sessions on all interfaces. Once a session has been terminated, the TOE requires the user to re-authenticate to establish a new session. The TOE displays an administratively configured banner on the local CLI, remote CLI, and Web GUI prior to allowing any administrative access to the TOE.

2.5.7 Trusted Path/Channels

The TOE connects and sends data to IT entities that reside in the Operational Environment via trusted channels. In the evaluated configuration, the TOE connects to an audit server using TLS to encrypt the audit data that traverses the channel and connects to the GigaVUE appliances using HTTPS. When accessing the TOE remotely, Security Administrators interface with the TSF using a trusted path. The remote CLI is protected via SSH and the Web GUI is protected via HTTPS.

3 Conformance Claims

3.1 CC Version

This ST is compliant with Common Criteria for Information Technology Security Evaluation, Version 3.1 Revision 5 April 2017.

3.2 CC Part 2 Conformance Claims

This ST and Target of Evaluation (TOE) is Part 2 extended to include all applicable NIAP and International interpretations through December 28, 2024.

3.3 CC Part 3 Conformance Claims

This ST and Target of Evaluation (TOE) are conformant to Part 3 to include all applicable NIAP and International interpretations through December 28, 2024.

3.4 PP Claims

This ST claims exact conformance to the following Protection Profiles:

- collaborative Protection Profile for Network Devices Version 2.2e [NDcPP]

3.5 Package Claims

The TOE claims exact conformance to the NDcPP, which is conformant with CC Part 3.

The TOE claims following Selection-Based SFRs that are defined in the appendices of the claimed PP:

- FCS_HTTPS_EXT.1
- FCS_SSHS_EXT.1
- FCS_TLSC_EXT.1
- FCS_TLSS_EXT.1
- FIA_X509_EXT.1/Rev
- FIA_X509_EXT.2
- FIA_X509_EXT.3
- FMT_MTD.1/CryptoKeys

The TOE claims the following Optional SFRs that are defined in the appendices of the claimed PP:

- FAU_STG.1

This does not violate the notion of exact conformance because the NDcPP specifically indicates these as allowable selections and options and provides both the ST author and evaluation laboratory with instructions on how these claims are to be documented and evaluated.

3.6 Package Name Conformant or Package Name Augmented

This ST and TOE are in exact conformance with the NDcPP.

3.7 Conformance Claim Rationale

The NDcPP states the following: "This is a Collaborative Protection Profile (cPP) whose Target of Evaluation (TOE) is a Network Device (ND)… A network device in the context of this cPP is a device connected to the network and has an infrastructure within the network… Examples of network devices that are covered by requirements in this cPP include physical and virtualized routers, firewalls, VPN gateways, IDSs, and switches."

The TOE is a network device composed of hardware and software that is designed to configure and enforce traffic flow policies as well as to provide visibility into the enterprise's network. As such, it can be understood as having a role in network infrastructure. Therefore, the conformance claim is appropriate.

3.8 Technical Decisions

TD #	Title	References	Changes			Analysis to this evaluation	
			SFR	AA	Notes	NA	Reason
TD0527	Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	FIA_X509_EXT.1/REV, FIA_X509_EXT.1/ITT		X			AA: Testing Update. No ST updates required.
TD0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	FCS_NTP_EXT.1.4, ND SD v2.2.		X		X	AA: Testing Update. SFR not claimed.
TD0536	NIT Technical Decision for Update Verification Inconsistency	AGD_OPE.1, ND SDv2.2.		X			AA: Guidance Update. No ST updates required.
TD0537	NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	FIA_X509_EXT.2.2			X		SFR claimed but Note change has no impact on evaluation documentation.
TD0546	NIT Technical Decision for DTLS - clarification of Application Note 63	FCS_DTLSC_EXT.1.1			X	X	SFR not claimed
TD0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	ND SDv2.2, AVA_VAN.1		X			Clarification of AVA_VAN No ST updates required.
TD0555	NIT Technical Decision for RFC Reference incorrect in TLSS Test	NDSDv2.2, FCS_TLSS_EXT.1.4, Test 3		X			AA: Testing update. No ST updates required.
TD0556	NIT Technical Decision for RFC 5077 question	NDSDv2.2, FCS_TLSS_EXT.1.4, Test 3		X			AA: Testing update. No ST updates required.
TD0563	NiT Technical Decision for Clarification of audit date information	NDcPPv2.2e, FAU_GEN.1.2			X		Clarified date time stamp requirements. No ST updates required. AGD Section 8 shows compliance.
TD0564	NiT Technical Decision for Vulnerability Analysis Search Criteria	NDSDv2.2, AVA_VAN.1			X		Clarified AVA public search requirements.

TD0569	NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	ND SD v2.2, FCS_DTLSS_EXT.1.7, FCS_TLSS_EXT.1.4		X	X		AA: Test No ST updates required.
TD0570	NiT Technical Decision for Clarification about FIA_AFL.1	FIA_AFL.1			X		Makes FIA_AFL.1 mandatory. FIA_AFL.1 was already claimed. Not marked with footnote as no SFR wording changes were mandated.
TD0571	NiT Technical Decision for Guidance on how to handle FIA_AFL.1	FIA_UAU.1, FIA_PMG_EXT.1			X		Makes FIA_PMG_EXT.1, FIA_AFL.1, and FMT_SMF.1 mandatory. All were previously claimed. Not marked with footnote as no SFR wording changes were mandated.
TD0572	NiT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	FTP_ITC.1			X		Clarification: no changes to AA or ST required.
TD0580	NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	FCS_CKM.1.1, FCS_CKM.2.1	X	X	X	X	Not claiming DH14 AA: TSS, Test
TD0581	NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	FCS_CKM.2	X				Updated revisions 2 to 3 Footnote 2
TD0591	NIT Technical Decision for Virtual TOEs and hypervisors	A.LIMITED_FUNCTIONALITY, ACRONYMS			X		A.LIMITED_FUNCTIONALITY wording change. Footnote 1
TD0592	NIT Technical Decision for Local Storage of Audit Records	FAU_STG			X		No changes to ST or AA activities. Only changes wording in the PP.
TD0631	NIT Technical Decision for Clarification of public key authentication for SSH Server	ND SDv2.2, FCS_SSHS_EXT.1, FMT_SMF.1	X	X	X		AA: TSS and Tests Footnote 3 for FCS_SSHS_EXT.1.2 Footnote 4 for FMT_SMF.1
TD0632	NIT Technical Decision for Consistency with Time Data for vNDs	ND SD2.2, FPT_STM_EXT.1.2	X	X	X	X	AA: TSS, AGD, Test Adds an SFR selection not being claimed. TOE is not vND.
TD0635	NIT Technical Decision for TLS Server and Key Agreement Parameters	FCS_TLSS_EXT.1.3, NDSD v2.2		X			AA: TSS No ST updates required.
TD0636	NIT Technical Decision for Clarification of Public Key User Authentication for SSH	ND SD2.2, FCS_SSHC_EXT.1	X	X	X	X	AA: TSS and Tests SSHC is not being claimed.

TD0638	NIT Technical Decision for Key Pair Generation for Authentication	NDSDv2.2, FCS_CKM.1			X		Update to PP footnote 4. No change to ST required.
TD0639	NIT Technical Decision for Clarification for NTP MAC Keys	FCS_NTP_EXT.1.2, FAU_GEN.1, FCS_CKM.4, FPT_SKP_EXT.1			X	X	The TOE is not claiming NTP usage
TD0670	NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	ND SD2.2, FCS_TLSC_EXT.2.1		X		X	AA: Tests Mutual authentication is not claimed
TD0738	NIT Technical Decision for Link to Allowed-With List	Chapter 2			X		PP claimed but Note change has no impact on evaluation documentation.
TD0790	NIT Technical Decision: Clarification Required for testing IPv6	FCS_DTLSC_EXT.1.2, FCS_TLSC_EXT1.2, CPP_ND_V2.2-SD		X			AA: Test No ST updates required
TD0792	NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR	FIA_PMG_EXT.1, CPP_ND_V2.2-SD		X			AA: TSS No ST updates required
TD0800	Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	FCS_IPSEC_EXT.1.7, FCS_IPSEC_EXT.1.8, CPP_ND_V2.2-SD		X		X	AA: AGD, Test Not claiming IPsec

4 Security Problem Definition

4.1 Threats

This section identifies the threats against the TOE. These threats have been taken from the NDcPP.

Threat	Threat Definition
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., 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

Threat	Threat Definition
	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.

Table 4-1: TOE Threats

4.2 Organizational Security Policies

This section identifies the organizational security policies which are expected to be implemented by an organization that deploys the TOE. These policies have been taken from the NDcPP.

Policy	Policy Definition
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.

Table 4-2: Organizational Security Policies

4.3 Assumptions

The specific conditions listed in this section are assumed to exist in the TOE's Operational Environment. These assumptions have been taken from the NDcPP.

Assumption	Assumption Definition
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.
A.LIMITED_FUNCTIONALITY[1]	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).

[1] TD0591

Assumption	Assumption Definition
	If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform.
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 ND cPP. 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.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.

Table 4-3: TOE Assumptions

4.4 Security Objectives

This section identifies the security objectives of the TOE and its supporting environment. The security objectives identify the responsibilities of the TOE and its environment in meeting the security needs.

4.4.1 TOE Security Objectives

The NDcPP does not define any security objectives for the TOE.

4.4.2 Security Objectives for the Operational Environment

The TOE's operational environment must satisfy the following objectives:

Objective	Objective Definition
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. Note: For vNDs the TOE includes only the contents of the its own VM, and does not include other VMs or the VS.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For vNDs, this includes the VS Administrator responsible for configuring the VMs that implement ND functionality. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment. For vNDs, this applies when the physical platform on which the VM runs is removed from its operational environment.

Table 4-4: Operational Environment Objectives

4.5 Security Problem Definition Rationale

The assumptions, threats, OSPs, and objectives that are defined in this ST represent the assumptions, threats, OSPs, and objectives that are specified in the Protection Profile to which the TOE claims conformance.

5 Extended Components Definition

5.1 Extended Security Functional Requirements

The extended Security Functional Requirements that are claimed in this ST are taken directly from the PP to which the ST and TOE claim conformance. These extended components are formally defined in the PP in which their usage is required.

5.2 Extended Security Assurance Requirements

The extended Security Assurance Requirements that are claimed in this ST are taken directly from the PP to which the ST and TOE claim conformance. These extended components are formally defined in the PP in which their usage is required.

6 Security Functional Requirements

6.1 Conventions

The CC permits four functional component operations—assignment, refinement, selection, and iteration—to be performed on functional requirements. This ST will highlight the operations in the following manner:

- **Assignment:** allows the specification of an identified parameter. Indicated with *italicized* text.
- **Refinement:** allows the addition of details. Indicated with **bold** text. Note that conversion of British spelling to American spelling is not marked as a refinement (e.g., 'authorisation' changed to 'authorization').
- **Selection:** allows the specification of one or more elements from a list. Indicated with underlined text.
- **Iteration:** allows a component to be used more than once with varying operations. Indicated with a sequential number in parentheses following the element number of the iterated SFR and/or separated by a "/" with a notation that references the function for which the iteration is used, e.g., "/TrustedUpdate" for an SFR that relates to update functionality

When multiple operations are combined, such as an assignment that is provided as an option within a selection or refinement, a combination of the text formatting is used.

If SFR text is reproduced verbatim from text that was formatted in a claimed PP (such as if the PP's instantiation of the SFR has a refinement or a completed assignment), the formatting is not preserved. This is so that the reader can identify the operations that are performed by the ST author as opposed to the PP author.

6.2 Security Functional Requirements Summary

The following table lists the SFRs claimed by the TOE:

Class Name	Component Identification	Component Name
Security Audit (FAU)	FAU_GEN.1	Audit Data Generation
	FAU_GEN.2	User Identity Association
	FAU_STG.1	Protected Audit Trail Storage
	FAU_STG_EXT.1	Protected Audit Event Storage
Cryptographic Support (FCS)	FCS_CKM.1	Cryptographic Key Generation
	FCS_CKM.2	Cryptographic Key Establishment
	FCS_CKM.4	Cryptographic Key Destruction
	FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
	FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
	FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
	FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
	FCS_RBG_EXT.1	Cryptographic Operation (Random Bit Generation)
	FCS_HTTPS_EXT.1	HTTPS Protocol

Class Name	Component Identification	Component Name
	FCS_SSHS_EXT.1	SSH Server Protocol
	FCS_TLSC_EXT.1	TLS Client Protocol Without Mutual Authentication
	FCS_TLSS_EXT.1	TLS Server Protocol Without Mutual Authentication
Identification and Authentication (FIA)	FIA_AFL.1	Authentication Failure Management
	FIA_PMG_EXT.1	Password Management
	FIA_UAU_EXT.2	Password-based Authentication Mechanism
	FIA_UAU.7	Protected Authentication Feedback
	FIA_UIA_EXT.1	User Identification and Authentication
	FIA_X509_EXT.1/Rev	X.509 Certificate Validation
	FIA_X509_EXT.2	X509 Certificate Authentication
	FIA_X509_EXT.3	X.509 Certificate Requests
Security Management (FMT)	FMT_MOF.1/ManualUpdate	Management of Security Functions Behavior
	FMT_MTD.1/CoreData	Management of TSF Data
	FMT_MTD.1/CryptoKeys	Management of TSF Data
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.2	Restrictions on Security Roles
Protection of the TSF (FPT)	FPT_APW_EXT.1	Protection of Administrator Passwords
	FPT_SKP_EXT.1	Protection of TSF Data (For Reading of All Pre-shared, Symmetric and Private Keys)
	FPT_STM_EXT.1	Reliable Time Stamps
	FPT_TST_EXT.1	TSF Testing
	FPT_TUD_EXT.1	Trusted Update
TOE Access (FTA)	FTA_SSL_EXT.1	TSF-initiated Session Locking
	FTA_SSL.3	TSF-Initiated Termination
	FTA_SSL.4	User-initiated Termination
	FTA_TAB.1	TOE Access Banners
Trusted Path/Channels (FTP)	FTP_ITC.1	Inter-TSF Trusted Channel
	FTP_TRP.1/Admin	Trusted Path

Table 6-1: Security Functional Requirements for the TOE

6.3 Security Functional Requirements

6.3.1 **Class FAU: Security Audit**

6.3.1.1 ***FAU_GEN.1 Audit Data Generation***

FAU_GEN.1.1

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions comprising:
    - Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).

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

d) Specifically defined auditable events listed in Table **6-2**.

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 **6-2**.

Requirement	Auditable Event(s)	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG.1	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure.
FCS_SSHS_EXT.1	Failure to establish an SSH session.	Reason for failure.
FCS_TLSC_EXT.1	Failure to establish a TLS session.	Reason for failure.
FCS_TLSS_EXT.1	Failure to establish a TLS session.	Reason for failure.
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP Address).
FIA_PMG_EXT.1	None.	None.
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP Address).
FIA_UAU.7	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP Address).
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.
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update**.**	None.
FMT_MTD.1/CoreData	None.	None.

FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_SKP_EXT.1	None.	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).
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure).	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_SSL_EXT.1	The termination of a local session by the session locking mechanism.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.

Table 6-2: Auditable Events

6.3.1.2 *FAU_GEN.2 User Identity Association*

FAU_GEN.2.1

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

6.3.1.3 *FAU_STG.1 Protected Audit Trail Storage*

FAU_STG.1.1

The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2

The TSF shall be able to prevent unauthorized modifications to the stored audit records in the audit trail.

6.3.1.4 *FAU_STG_EXT.1 Protected Audit Event Storage*

FAU_STG_EXT.1.1

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

FAU_STG_EXT.1.2

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

- The TOE shall consist of a single standalone component that stores audit data locally].

FAU_STG_EXT.1.3

The TSF shall [[*rotate compressed audit archived audit files on a First in First out (FIFO) basis according to the following rule:*

- *Delete oldest archived log file*
- *rotate remaining archived log files*
- *close, compress, and archive current log file*
- *open new audit log file to receive current entries*

]] when the local storage space for audit data is full.

6.3.2 Class FCS: Cryptographic Support

6.3.2.1 *FCS_CKM.1 Cryptographic Key Generation*

FCS_CKM.1.1

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

- 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].

6.3.2.2 *FCS_CKM.2 Cryptographic Key Establishment*

FCS_CKM.2.1[2]

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

- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"].

6.3.2.3 *FCS_CKM.4 Cryptographic Key Destruction*

FCS_CKM.4.1

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

[2] TD0581

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

that meets the following: No Standard.

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

FCS_COP.1.1/DataEncryption

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

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

FCS_COP.1.1/SigGen

The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [

- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [*256 bits*]]

that meet the following: [

- 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].

6.3.2.6 *FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)*

FCS_COP.1.1/Hash

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

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

FCS_COP.1.1/KeyedHash

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

6.3.2.8 *FCS_RBG_EXT.1 Cryptographic Operation (Random Bit Generation)*

FCS_RBG_EXT.1.1

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

FCS_RBG_EXT.1.2

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

6.3.2.9 *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.

6.3.2.10 *FCS_SSHS_EXT.1 SSH Server Protocol*

FCS_SSHS_EXT.1.1

The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [5656, 6668].

FCS_SSHS_EXT.1.2[3]

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

FCS_SSHS_EXT.1.3

The TSF shall ensure that, as described in RFC 4253, packets greater than [*32,768*] 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-ctr, aes256-ctr, aes128-gcm@openssh.com, aes256-gcm@openssh.com].

[3] TD0631

FCS_SSHS_EXT.1.5

The TSF shall ensure that the SSH public-key based authentication implementation uses [ecdh-sha2-nistp384] 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-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 [ecdh-sha2-nistp256] and [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.

6.3.2.11 *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_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]

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, and no other attribute types].

FCS_TLSC_EXT.1.3

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

- Not implement any administrator override mechanism

].

FCS_TLSC_EXT.1.4

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

6.3.2.12 *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_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]

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 [ECDHE curves [secp384r1] and no other curves].

FCS_TLSS_EXT.1.4

The TSF shall support [no session resumption or session tickets].

6.3.3 Class FIA: Identification and Authentication

6.3.3.1 *FIA_AFL.1 Authentication Failure Management*

FIA_AFL.1.1

The TSF shall detect when an Administrator configurable positive integer within [*3-5*] 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 [*a manual unlock of the account*] is taken by an Administrator, 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].

6.3.3.2 *FIA_PMG_EXT.1 Password Management*

FIA_PMG_EXT.1.1

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”];
b) Minimum password length shall be configurable to between [*8*] and [*64*] characters.

6.3.3.3 *FIA_UAU_EXT.2 Password-Based Authentication Mechanism*

FIA_UAU_EXT.2.1

The TSF shall provide a local [password-based, SSH public key-based] authentication mechanism to perform local administrative user authentication.

6.3.3.4 *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.

6.3.3.5 *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.

6.3.3.6 *FIA_X509_EXT.1/Rev X.509 Certificate Validation*

FIA_X509_EXT.1.1/Rev

The TSF shall validate certificates in accordance with the following rules:

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

- Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
- Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
- Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
- OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

FIA_X509_EXT.1.2/Rev

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

6.3.3.7 *FIA_X509_EXT.2 X.509 Certificate Authentication*

FIA_X509_EXT.2.1

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

FIA_X509_EXT.2.2

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

6.3.3.8 *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].

FIA_X509_EXT.3.2

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

6.3.4 Class FMT: Security Management

6.3.4.1 *FMT_MOF.1/ManualUpdate Management of Security Functions Behavior*

FMT_MOF.1.1/ManualUpdate

The TSF shall restrict the ability to enable the functions to perform manual updates to Security Administrators.

6.3.4.2 *FMT_MTD.1/CoreData Management of TSF Data*

FMT_MTD.1.1/CoreData

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

6.3.4.3 *FMT_MTD.1/CryptoKeys Management of TSF Data*

FMT_MTD.1.1/CryptoKeys

The TSF shall restrict the ability to manage the cryptographic keys to Security Administrators.

6.3.4.4 *FMT_SMF.1 Specification of Management Functions*

FMT_SMF.1.1[4]

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 [hash comparison] capability prior to installing those updates;
- Ability to configure the authentication failure parameters for FIA_AFL.1;
- [
  - Ability to manage the cryptographic keys;
  - Ability to configure thresholds for SSH rekeying;
  - Ability to set the time which is used for time-stamps;
  - Ability to re-enable an Administrator account;
  - 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].

6.3.4.5 *FMT_SMR.2 Restrictions on Security Roles*

FMT_SMR.2.1

The TSF shall maintain the roles:

- Security Administrator.

FMT_SMR.2.2

The TSF shall be able to associate users with roles.

FMT_SMR.2.3

The TSF shall ensure that the conditions

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

[4] TD0631

are satisfied.

6.3.5 Class FPT: Protection of the TSF

6.3.5.1 *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.

6.3.5.2 *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.

6.3.5.3 *FPT_STM_EXT.1 Reliable Time Stamps*

FPT_STM_EXT.1.1

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

FPT_STM_EXT.1.2

The TSF shall [allow the Security Administrator to set the time].

6.3.5.4 *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), at the request of the authorized user] to demonstrate the correct operation of the TSF: [*hardware checks, filesystem integrity check, validation of cryptographic functions, and TOE software integrity check*].

6.3.5.5 *FPT_TUD_EXT.1 Trusted Update*

FPT_TUD_EXT.1.1

The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software and [the most recently installed version of the TOE firmware/software].

FPT_TUD_EXT.1.2

The TSF shall provide Security Administrators the ability to manually initiate updates to TOE firmware/software and [no other update mechanism].

FPT_TUD_EXT.1.3

The TSF shall provide means to authenticate firmware/software updates to the TOE using a [published hash] prior to installing those updates.

6.3.6 Class FTA: TOE Access

6.3.6.1 *FTA_SSL_EXT.1 TSF-Initiated Session Locking*

FTA_SSL_EXT.1.1

The TSF shall, for local interactive sessions, [

- terminate the session]

after a Security Administrator-specified time period of inactivity.

6.3.6.2 *FTA_SSL.3 TSF-Initiated Termination*

FTA_SSL.3.1

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

6.3.6.3 *FTA_SSL.4 User-Initiated Termination*

FTA_SSL.4.1

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

6.3.6.4 *FTA_TAB.1 TOE Access Banner*

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.

6.3.7 Class FTP: Trusted Path/Channels

6.3.7.1 *FTP_ITC.1 Inter-TSF Trusted Channel*

FTP_ITC.1.1

The TSF shall be capable of using [TLS, HTTPS] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [[*GigaVUE appliances*]] 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 [*export audit to audit server, communication to GigaVUE appliances*].

6.3.7.2 *FTP_TRP.1/Admin Trusted Path*

FTP_TRP.1.1/Admin

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

FTP_TRP.1.2/Admin

The TSF shall permit remote Administrators to initiate communication via the trusted path.

FTP_TRP.1.3/Admin

The TSF shall require the use of the trusted path for initial Administrator authentication and all remote administration actions.

6.4 Statement of Security Functional Requirements Consistency

The Security Functional Requirements included in the ST represent all required SFRs specified in the claimed PP, a subset of the optional requirements, and all applicable selection-based requirements that have been included as specified for the claimed PP.

7 Security Assurance Requirements

This section identifies the Security Assurance Requirements (SARs) that are claimed for the TOE. The SARs which are claimed are in exact conformance with the NDcPP.

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

7.1 Class ASE: Security Target evaluation

7.1.1 ST introduction (ASE_INT.1)

7.1.1.1 *Developer action elements:*

ASE_INT.1.1D

The developer shall provide an ST introduction.

7.1.1.2 *Content and presentation elements:*

ASE_INT.1.1C

The ST introduction shall contain an ST reference, a TOE reference, a TOE overview and a TOE description.

ASE_INT.1.2C

The ST reference shall uniquely identify the ST.

ASE_INT.1.3C

The TOE reference shall uniquely identify the TOE.

ASE_INT.1.4C

The TOE overview shall summarize the usage and major security features of the TOE.

ASE_INT.1.5C

The TOE overview shall identify the TOE type.

ASE_INT.1.6C

The TOE overview shall identify any non-TOE hardware/software/firmware required by the TOE.

ASE_INT.1.7C

The TOE description shall describe the physical scope of the TOE.

ASE_INT.1.8C

The TOE description shall describe the logical scope of the TOE.

7.1.1.3 *Evaluator action elements:*

ASE_INT.1.1E

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

ASE_INT.1.2E

The evaluator shall confirm that the TOE reference, the TOE overview, and the TOE description are consistent with each other.

7.1.2 Conformance claims (ASE_CCL.1)

7.1.2.1 *Developer action elements:*

ASE_CCL.1.1D

The developer shall provide a conformance claim.

ASE_CCL.1.2D

The developer shall provide a conformance claim rationale

7.1.2.2 *Content and presentation elements:*

ASE_CCL.1.1C

The conformance claim shall contain a CC conformance claim that identifies the version of the CC to which the ST and the TOE claim conformance.

ASE_CCL.1.2C

The CC conformance claim shall describe the conformance of the ST to CC Part 2 as either CC Part 2 conformant or CC Part 2 extended.

ASE_CCL.1.3C

The CC conformance claim shall describe the conformance of the ST to CC Part 3 as either CC Part 3 conformant or CC Part 3 extended.

ASE_CCL.1.4C

The CC conformance claim shall be consistent with the extended components definition.

ASE_CCL.1.5C

The conformance claim shall identify all PPs and security requirement packages to which the ST claims conformance.

ASE_CCL.1.6C

The conformance claim shall describe any conformance of the ST to a package as either package-conformant or package-augmented.

ASE_CCL.1.7C

The conformance claim rationale shall demonstrate that the TOE type is consistent with the TOE type in the PPs for which conformance is being claimed.

ASE_CCL.1.8C

The conformance claim rationale shall demonstrate that the statement of the security problem definition is consistent with the statement of the security problem definition in the PPs for which conformance is being claimed.

ASE_CCL.1.9C

The conformance claim rationale shall demonstrate that the statement of security objectives is consistent with the statement of security objectives in the PPs for which conformance is being claimed.

ASE_CCL.1.10C

The conformance claim rationale shall demonstrate that the statement of security requirements is consistent with the statement of security requirements in the PPs for which conformance is being claimed.

7.1.2.3 *Evaluator action elements:*

ASE_CCL.1.1E

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

7.1.3 Security problem definition (ASE_SPD)

7.1.3.1 *Developer action elements:*

ASE_SPD.1.1D

The developer shall provide a security problem definition.

7.1.3.2 *Content and presentation elements:*

ASE_SPD.1.1C

The security problem definition shall describe the threats.

ASE_SPD.1.2C

All threats shall be described in terms of a threat agent, an asset, and an adverse action.

ASE_SPD.1.3C

The security problem definition shall describe the OSPs.

ASE_SPD.1.4C

The security problem definition shall describe the assumptions about the operational environment of the TOE.

7.1.3.3 *Evaluator action elements:*

ASE_SPD.1.1E

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

7.1.4 Security objectives for the operational environment (ASE_OBJ.1)

7.1.4.1 *Developer action elements:*

ASE_OBJ.1.1D

The developer shall provide a statement of security objectives.

7.1.4.2 *Content and presentation elements:*

ASE_OBJ.1.1C

The statement of security objectives shall describe the security objectives for the operational environment.

7.1.4.3 *Evaluator action elements:*

ASE_OBJ.1.1E

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

7.1.5 Extended components definition (ASE_ECD.1)

7.1.5.1 *Developer action elements:*

ASE_ECD.1.1D

The developer shall provide a statement of security requirements.

ASE_ECD.1.2D

The developer shall provide an extended components definition.

7.1.5.2 *Content and presentation elements:*

ASE_ECD.1.1C

The statement of security requirements shall identify all extended security requirements.

ASE_ECD.1.2C

The extended components definition shall define an extended component for each extended security requirement.

ASE_ECD.1.3C

The extended components definition shall describe how each extended component is related to the existing CC components, families, and classes.

ASE_ECD.1.4C

The extended components definition shall use the existing CC components, families, classes, and methodology as a model for presentation.

ASE_ECD.1.5C

The extended components shall consist of measurable and objective elements such that conformance or nonconformance to these elements can be demonstrated.

7.1.5.3 *Evaluator action elements:*

ASE_ECD.1.1E

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

ASE_ECD.1.2E

The evaluator shall confirm that no extended component can be clearly expressed using existing components.

7.1.6 Stated security requirements (ASE_REQ.1)

7.1.6.1 *Developer action elements:*

ASE_REQ.1.1D

The developer shall provide a statement of security requirements.

ASE_REQ.1.2D

The developer shall provide a security requirements rationale.

7.1.6.2 *Content and presentation elements:*

ASE_REQ.1.1C

The statement of security requirements shall describe the SFRs and the SARs.

ASE_REQ.1.2C

All subjects, objects, operations, security attributes, external entities and other terms that are used in the SFRs and the SARs shall be defined.

ASE_REQ.1.3C

The statement of security requirements shall identify all operations on the security requirements.

ASE_REQ.1.4C

All operations shall be performed correctly.

ASE_REQ.1.5C

Each dependency of the security requirements shall either be satisfied, or the security requirements rationale shall justify the dependency not being satisfied.

ASE_REQ.1.6C

The statement of security requirements shall be internally consistent.

7.1.6.3 *Evaluator action elements:*

ASE_REQ.1.1E

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

7.1.7 TOE summary specification (ASE_TSS.1)

7.1.7.1 *Developer action elements:*

ASE_TSS.1.1D

The developer shall provide a TOE summary specification.

7.1.7.2 *Content and presentation elements:*

ASE_TSS.1.1C

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.

7.1.7.3 *Evaluator action elements:*

ASE_TSS.1.1E

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

ASE_TSS.1.2E

The evaluator shall confirm that the TOE summary specification is consistent with the TOE overview and the TOE description.

7.2 Class ADV: Development

7.2.1 **Basic Functional Specification (ADV_FSP.1)**

7.2.1.1 ***Developer action elements:***

ADV_FSP.1.1D

The developer shall provide a functional specification.

ADV_FSP.1.2D

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

7.2.1.2 ***Content and presentation elements:***

ADV_FSP.1.1C

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

ADV_FSP.1.2C

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

ADV_FSP.1.3C

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

ADV_FSP.1.4C

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

7.2.1.3 ***Evaluator action elements:***

ADV_ FSP.1.1E

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

ADV_ FSP.1.2E

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

7.3 Class AGD: Guidance Documentation

7.3.1 Operational User Guidance (AGD_OPE.1)

7.3.1.1 *Developer action elements:*

AGD_OPE.1.1D

The developer shall provide operational user guidance.

7.3.1.2 *Content and presentation elements:*

AGD_OPE.1.1C

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

AGD_OPE.1.2C

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

AGD_OPE.1.3C

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

AGD_OPE.1.4C

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

AGD_OPE.1.5C

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

AGD_OPE.1.6C

The operational user guidance shall, for each user role, describe the security measures to be followed in order to fulfill the security objectives for the operational environment as described in the ST.

AGD_OPE.1.7C

The operational user guidance shall be clear and reasonable.

7.3.1.3 *Evaluator action elements:*

AGD_OPE.1.1E

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

7.3.2 Preparative Procedures (AGD_PRE.1)

7.3.2.1 *Developer action elements:*

AGD_PRE.1.1D

The developer shall provide the TOE including its preparative procedures.

7.3.2.2 *Content and presentation elements:*

AGD_ PRE.1.1C

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

AGD_ PRE.1.2C

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

7.3.2.3 *Evaluator action elements:*

AGD_ PRE.1.1E

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

AGD_ PRE.1.2E

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

7.4 Class ALC: Life Cycle Supports

7.4.1 Labeling of the TOE (ALC_CMC.1)

7.4.1.1 *Developer action elements:*

ALC_CMC.1.1D

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

7.4.1.2 *Content and presentation elements:*

ALC_CMC.1.1C

The TOE shall be labeled with its unique reference.

7.4.1.3 ***Evaluator action elements:***

ALC_CMC.1.1E

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

7.4.2 TOE CM Coverage (ALC_CMS.1)

7.4.2.1 ***Developer action elements:***

ALC_CMS.1.1D

The developer shall provide a configuration list for the TOE.

7.4.2.2 ***Content and presentation elements:***

ALC_CMS.1.1C

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

ALC_CMS.1.2C

The configuration list shall uniquely identify the configuration items.

7.4.2.3 ***Evaluator action elements:***

ALC_CMS.1.1E

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

7.5 Class ATE: Tests

7.5.1 Independent Testing - Conformance (ATE_IND.1)

7.5.1.1 ***Developer action elements:***

ATE_IND.1.1D

The developer shall provide the TOE for testing.

7.5.1.2 ***Content and presentation elements:***

ATE_IND.1.1C

The TOE shall be suitable for testing.

7.5.1.3 ***Evaluator action elements:***

ATE_IND.1.1E

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

ATE_IND.1.2E

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

7.6 Class AVA: Vulnerability Assessment

7.6.1 **Vulnerability Survey (AVA_VAN.1)**

7.6.1.1 ***Developer action elements:***

AVA_VAN.1.1D

The developer shall provide the TOE for testing.

7.6.1.2 ***Content and presentation elements:***

AVA_VAN.1.1C

The TOE shall be suitable for testing.

7.6.1.3 ***Evaluator action elements:***

AVA_VAN.1.1E

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

AVA_VAN.1.2E

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

AVA_VAN.1.3E

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

8 TOE Summary Specification

The following sections identify the security functions of the TOE and describe how the TSF meets each claimed SFR. They include Security Audit, Cryptographic Support, Identification and Authentication, Security Management, Protection of the TSF, TOE Access, and Trusted Path/Channels.

8.1 Security Audit

8.1.1 **FAU_GEN.1:**

The TOE contains mechanisms to generate audit data based upon successful and unsuccessful management actions performed by all authorized users of the TOE. Each audit record contains identifying information of the subject performing the action. The audit records are generated and stored in the form of syslog records which are sent securely to the audit server protected by TLS. The TOE allows for a Security Administrator to view audit records through the Local CLI, Remote CLI, and Web GUI.

The auditable events include the start-up and shut-down of the audit functions; administrative actions including login, logout, TSF configuration changes, managing cryptographic keys and resetting passwords; and all events defined in Table 6-2. Audit records for cryptographic functions, such as generating/import of, changing, or deleting cryptographic keys, will contain the value that represents the key to identify the key. The audit records that the TOE creates include the following information: date and time of the event (year, month, day, hour, minute, sec), event type, subject identity, and success or failure of the event. Additionally, specific events require additional information as defined by the 'Additional Audit Record Contents' column of Table 6-2.

8.1.2 **FAU_GEN.2:**

The TOE records the identity of the user (e.g., username, system name, IP address) associated with each audited event in the audit record.

8.1.3 **FAU_STG.1:**

Audit records can be viewed using any of the Security Administrator interfaces. There is no access to delete or modify audit records through the Web GUI. The audit log files can be accessed via the CLI by a Security Administrator that has the ability to escalate to root privileges, using the sudo command, to make authorized file deletions or modifications.

The amount of audit data stored locally on the TOE is described in Section 8.1.4.

8.1.4 **FAU_STG_EXT.1:**

In the evaluated configuration, the TOE will send audit records to a remote audit server through a TLS trusted channel. The audit records are stored locally on the TOE and immediately pushed to an audit server in the operational environment. If the audit server connectivity from the TOE is unavailable, audit records will only be stored locally. Upon re-establishment of communications between the TOE and the audit server, new audit records are transmitted. Any audit records generated during the time the audit server connection was down remain in storage locally and are not sent to the audit server. This is a standalone TOE that is responsible for storing and sending its own generated audit records.

The TOE stores one current audit log file and up to ten archived audit log files. The maximum capacity of a single audit log file is 10MB, and the total capacity is a maximum of 110MB. Audit log files perform file rotation on a First in First out (FIFO) basis which is triggered based upon the current audit log file needing to be archived and there are already ten archived files present on the TOE. The active audit log file is archived when it reaches 10MB or when the calendar date changes. When the file rotation sequence is triggered, the oldest log file is deleted from the archive to free up space for the rotation. The remaining log files are rotated and renamed. The currently open log file is closed, compressed, and archived. Finally, a new audit log file is created to store new entries.

The TOE protects audit records from unauthorized modification and deletion by limiting access to the CLI to Security Administrators only and by not providing functionality to modify or delete audit records through the Web GUI. A Security Administrator can view audit records through the CLI or the Web GUI.

8.2 Cryptographic Support

The TOE implements the Bouncy Castle version 1.0.2.3 cryptographic library. The Bouncy Castle library include algorithms that are certified under the following consolidated CAVP certificates:

a) Bouncy Castle library under CAVP Certificate #A6377

The following tables contain the CAVP algorithm certificates for the cryptographic library implemented in the TOE:

SFR	Algorithm Cert	CAVP Cert #
FCS_CKM.1- ECC schemes	ECDSA KeyGen (FIPS186-4) P-256, P-384, and P-521 ECDSA KeyVer (FIPS186-4) P-256, P-384, and P-521	#A6377
FCS_CKM.2 - ECDSA	KAS-ECC-SSC Sp800-56Ar3	#A6377
FCS_COP.1/DataEncryption	AES CBC 128 bits and 256 bits AES CTR 128 bits and 256 bits AES GCM 128 bits and 256 bits	#A6377
FCS_COP.1/SigGen	ECDSA FIPS 186-4 Signature Services 256 bits, NIST P-256, P-384, and P-521 curves	#A6377
FCS_COP.1/Hash	SHA-256, SHA-384, and SHA-512	#A6377
FCS_COP.1/KeyedHash	HMAC-256, HMAC-384, and HMAC-512	#A6377
FCS_RBG_EXT.1	Hash_DRBG	#A6377

Table 8-1: Cryptographic Algorithm Table for Bouncy Castle

8.2.1 **FCS_CKM.1:**

The TOE generates ECC keys using NIST curve P-256, P-384, and P-521, in accordance with FIPS PUB 186-4. The ECC keys are generated in support of device authentication for TLS and SSH.

The TOE's key generation function is validated under CAVP ECDSA certificate: #A6377.

8.2.2 **FCS_CKM.2:**

The TOE implements NIST SP 800-56A Revision 3 conformant key establishment mechanisms for Elliptic Curve Diffie-Hellman (ECDH) key establishment schemes. Specifically, the TOE complies with

the NIST SP 800-56A Revision 3 key agreement scheme (KAS) primitives that are defined in section 5.6 of the SP. This is used for the establishment of TLS sessions for which the TOE can act as a TLS client and TLS server, and SSH sessions for which the TOE can act as a SSH server.

The TOE's implementation of NIST SP 800-56A is validated under CAVP KAS-SSC ECC certificate #A6377.

The following table provides an overview of the usage for each scheme:

Scheme	SFR	Service
ECDH	FCS_TLSC_EXT.1.1	Audit server connection
ECDH	FCS_TLSC_EXT.1.1 (FCS_HTTPS_EXT.1)	GigaVUE appliance connection
ECDH	FCS_TLSS_EXT.1.1 (FCS_HTTPS_EXT.1)	Web GUI administration
ECDH	FCS_SSHS_EXT.1.7	CLI administration

Table 8-2: Cryptographic Key Establishment Scheme Usage

8.2.3 **FCS_CKM.4:**

The TOE implements secure key destruction according to table 8-3. Table 8-3 describes what keys were used, where they are stored, and also how they are destroyed. The TOE is not subject to any situations that would prevent or delay key destruction and strictly conforms to the key destruction requirements. This combined approach protects the keys in volatile and non-volatile memory from being compromised. The following table identifies the keys and CSPs that are applicable to the TOE as well as their usage, storage location, and method of destruction:

Name	Origin	Store	Zeroization / Destruction
ecdsa-sha2-nistp256 **ecdsa-sha2-nistp384** **ecdsa-sha2-nistp521**	SSH Server application	RAM	Destroyed by a single direct overwrite consisting of zeroes (0x00). The key is zeroized immediately after it is no longer needed and when the TOE is shutdown or reinitialized. Automatically zeroized after exchange. From openSSH: secure_memzero()
ECDSA private key	SSH Server application	RAM	Destroyed by a single direct overwrite consisting of zeroes (0x00). The key is zeroized immediately after it is no longer needed and when the TOE is shutdown or reinitialized. Automatically zeroized after exchange From openSSH: secure_memzero()
SSH session keys	SSH Server application	RAM	Destroyed by a single direct overwrite consisting of zeroes (0x00).

Name	Origin	Store	Zeroization / Destruction
			The key is zeroized immediately after it is no longer needed and when the TOE is shutdown or reinitialized. Automatically zeroized after SSH session is terminated. From openSSH: secure_memzero()
SSH Server Host Private Key	Generated on platform during initial setup of device.	Filesystem	The Security Administrator destroys this key via the CLI by entering a command that will delete the key. When the TOE processes this command, it destroys the abstraction that represented the key. The Security Administrator would perform this action when they want to replace the key.
TLS Server Host Certificate Private Key (X.509 Certificate)	Generated on platform during initial setup or imported after installation.	Filesystem	The Security Administrator destroys this key via the CLI by entering a command that will delete the key. When the TOE processes this command, it destroys the abstraction that represented the key. The Security Administrator would perform this action when they want to replace the key.
TLS session keys	Generated by TOE's TLS Server application	RAM	Destroyed by destruction of reference to the key directly followed by a request for garbage collection. The key is destroyed immediately after it is no longer needed and when the TOE is shutdown or reinitialized. Automatic destroyed after TLS session is terminated.
TLS session keys	Generated by TOE's TLS Client applications	RAM	Destroyed by a single direct overwrite consisting of zeroes (0x00). The key is zeroized immediately after it is no longer needed and when the TOE is shutdown or reinitialized. Automatic zeroized after TLS session is terminated.

Table 8-3: Cryptographic Materials, Storage, and Destruction Methods

8.2.4 **FCS_COP.1/DataEncryption:**

The TOE performs encryption and decryption using the AES algorithm in CBC, CTR, and GCM mode with key sizes of 128 and 256 bits. This algorithm implementation is validated under CAVP AES certificate #A6377. The AES algorithm meets ISO 18033-3, the CBC mode implementation meets ISO 10116, CTR- as specified in ISO 10116, and the GCM mode implementation meets ISO 19772.

8.2.5 **FCS_COP.1/SigGen:**

The TOE performs signature generation and validation using Elliptic Curve Digital Signature Algorithm (ECDSA). The TOE supports ECDSA with 256-bit key size and implements the NIST P-256, P-384, and P-521 curves. The ECDSA implementation meets ISO/IEC 14888-3 Section 6.4 and FIPS PUB 186-4. This implementation is validated under CAVP ECSDA certificate #A6377.

8.2.6 **FCS_COP.1/Hash:**

The TOE provides cryptographic hashing services using SHA-256, SHA-384, and SHA-512 with message digest sizes of 256, 384, and 512 bits respectively, as specified in ISO/IEC 10118-3:2004. The TSF uses hashing services the following functions:

- SHA-256, and SHA-512 for SSH data integrity
- SHA-256, and SHA-384 for TLS
- SHA-256, SHA-384, and SHA-512 for TLS NIST curves
- SHA-256, SHA-384, and SHA-512 for HMAC
- SHA-512 for software integrity
- SHA-512 for password hashing

The SHA algorithm meets ISO/IEC 10118-3:2004 and is validated under CAVP SHS certificate #A6377.

8.2.7 **FCS_COP.1/KeyedHash:**

The TOE provides keyed-hashing message authentication services using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512. The HMAC implementation supports key sizes that are equal to block sizes. HMAC is implemented as specified in ISO/IEC 9797-2:2011 Section 7 “MAC Algorithm 2”, and the following MAC sizes are supported:

- HMAC-SHA-256: [key size: 256 bits, digest size: 256 bits, block size: 256 bits, MAC output length: 256 bits] for SSH and TLS
- HMAC-SHA-384: [key size: 384 bits, digest size: 384 bits, block size: 384 bits, MAC output length: 384 bits] for TLS
- HMAC-SHA-512: [key size: 512 bits, digest size: 512 bits, block size: 512 bits, MAC output length: 512 bits] for SSH

When the TOE uses AES in GCM mode for SSH, the keyed-hashing message authentication is implicit through the selection of AES-GCM for the data integrity MAC algorithm.

The algorithm is validated under CAVP HMAC certificate #A6377.

8.2.8 **FCS_RBG_EXT.1:**

The TOE implements a NIST-Approved deterministic random bit generator (DRBG). The DRBG used by the TOE is the Hash_DRBG as specified by ISO/IEC 18031:2011. The TOE provides one platform based noise source as described in the proprietary entropy specification. The DRBG is seeded with a minimum of 256 bits of entropy; so that it is sufficient to ensure full entropy for 256-bit keys, which are the largest keys generated by the TSF. The TOE’s DRBG implementation meets ISO/IEC 18031:2011 and is validated under CAVP certificate #A6377.

8.2.9 **FCS_HTTPS_EXT.1:**

The TSF implements HTTPS in compliance with RFC 2818. HTTPS Server functionality is used for remote administration of the TOE via the Web GUI. If a peer certificate is presented to the TSF, the TSF will not require client authentication if it is deemed invalid. The TSF also implements HTTPS Client

functionality to communicate with the GigaVUE appliances. The following summarizes how the TOE conforms to RFC 2818.

Section 2.1 Connection Initiation:	The TOE acts as both the HTTPS and TLS Client when initiating a connection. Conformant to this section.
Section 2.2 Connection Closure:	The TOE sends TLS closure alert when terminating an HTTPS connection. The TOE does not support session reuse. The TOE meets the behavior as described, without deviation. Conformant to this section.
Section 2.2.1 Client Behavior:	The TOE operates as a HTTPS Client. The TOE send a closure alert before closing the connection. Conformant to this section.
Section 2.2.2 Server Behavior:	The TOE does not support session resumption. The TOE attempts to initiate an exchange of closure alerts with the client before closing the connection. Conformant to this section.
Section 2.3 Port Number:	The TOE utilizes TCP port 443 to listen for incoming HTTPS connections. Conformant to this section.
Section 2.4 URI Format:	The TOE supports and requires the https:// URI protocol identifier prefix for incoming HTTPS requests. Conformant to this section.
Section 3.1 Server Identity:	The TOE checks the hostname against the server's identity as presented in the server's Certificate message. The TOE prioritizes the use of the subjectAltName extension of type dNSName, for the Gigamon-FM to GigaVUE appliances. Conformant to this section.
Section 3.2 Client Identity:	The TOE does not support mutual authentication for the HTTPS Server interface.

8.2.10 **FCS_SSHS_EXT.1:**

The TOE uses SSHv2 to secure the remote CLI management connection (SSH server) to the TOE. The traffic for this connection is received by the TOE on the SSHv2 port 22. The TOE's implementation of SSHv2 protocol complies with the following RFCs: 4251, 4252, 4253, 4254, 5656, and 6668. The TOE supports password-based and public key-based user authentication methods as described in RFC 4252; both authentication methods are supported for the remote CLI. The TOE's SSH server implementation allows for the use of ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 for public key user authentication. The TOE verifies that the client's presented public key matches one that is stored within the authorized-keys file.

The TOE's SSH server implementation provides data encryption using the aes128-ctr, aes256-ctr, aes128-gcm@openssh.com, and aes256-gcm@openssh.com encryption algorithms. The TOE's SSH server implementation allows for the use of only ecdh-sha2-nistp384 for host public key authentication and will reject all others. The MAC algorithms used for the data integrity are HMAC-SHA2-256 and HMAC-SHA2-512. The MAC algorithm is also implicit due to the selection of aes128-gcm@openssh.com and aes256-gcm@openssh.com for encryption algorithms. The TOE rejects all other MAC algorithms. The

only key exchange methods used by the TOE with SSHv2 are ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521.

All SSHv2 connections will be dropped upon detection of any packet greater than 32,768 bytes being transported, as described in RFC 4253. Additionally, session keys are created when the TOE establishes an SSHv2 connection. The TOE will monitor the time period during which the SSHv2 session keys are active and how much data has been transmitted using them. The TOE initiates a rekey when the session keys have been used for no longer than 1 hour or when no more than 1GB of data has been transmitted, whichever threshold is reached first.

8.2.11 **FCS_TLSC_EXT.1/ FCS_TLSS_EXT.1:**

The TOE uses TLS 1.2 protocol for securing transport of audit records sent to the remote audit server (TLS Client), for securing the connection between the TOE and GigaVUE appliances (TLS Client) and for remote administration through the Web GUI (TLS Server). The TOE will reject all connection attempts from TLS versions other than 1.2. In the evaluated configuration, the TOE will use the following ciphersuites for TLS client and server functionality:

- 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

When the TOE uses TLS Client functionality, the presented identifier for the server certificate has to match the reference identifier in order to establish the connection, per RFC 6125 Section 6. The TLS Client communications is to connect to the audit server or the GigaVUE appliances, and the TOE advertises the support for NIST curves secp256r1, secp384r1, and secp521r1 based upon the configuration defined by the Security Administrator. The connection between the TOE and the GigaVUE appliances claims FQDNS names use for CN and SAN, and only supports the use of wildcards in the left most label only. The connection between the TOE and the audit server claims IPv4 address use for CN and mandates the use of SAN, and does not support the use of wildcards. Canonical formatting according to RFC 3986 is enforced.

The TSF converts that IP address, obtained from the certificate, from ASN.1 to the binary representation of the textual string of the IP address. The TSF also converts the IP address from the established network connection to the binary representation of the textual string of the IP address. The two representations are then compared to determine what action is performed next.

- If the SAN field is not used (non-existent):
  - The certificate is deemed invalid and the connection is immediately terminated.
- If the SAN value exists:
  - If the two values match, revocation checking using the CRL is performed.
  - If the two values do not match, the certificate is deemed invalid and the connection is immediately terminated.

When certificate validation fails for either of these connections, the connection is not established. There is no administrative override mechanism to force the connection if the peer certificate is deemed invalid.

The TOE provides TLS Server support for connections from the administrative workstation's browser, and the TOE only claims connection establishment support using NIST curve secp384r1. Neither session resumption nor session tickets are supported by the TOE.

8.3 Identification and Authentication

8.3.1 **FIA_AFL.1:**

The TSF provides a configurable counter for consecutive failed authentication attempts that will lock a Security Administrator or user account when the failure counter threshold is reached. The remote CLI and Web GUI have separate counters for consecutive failed authentication attempts. User accounts are separate between the Web GUI and CLI. Therefore, when an account is locked, the offending Security Administrator or user cannot login to the interface that the account lock was triggered by. For example, an account locked while attempting a connection to the remote CLI, will not be granted access to the CLI until the offending user account is unlocked. Likewise, a Web GUI user account that was locked while attempting access to the Web GUI, will not be granted access until the offending user account is unlocked. A valid login that happens prior to the failure counter reaching its threshold will reset the counter to zero.

While a user account is locked, no authentication is possible. The lockout duration, for the remote CLI is an administratively configurable number in seconds and is established by an Admin role. For the Web GUI, a locked account must be manually unlocked by a Super Admin role user. To prevent a situation in which there is no administrative access to the TOE, the "admin" user account for the local CLI is not subject to the account becoming locked after the maximum number of authentication attempts is reached.

8.3.2 **FIA_PMG_EXT.1:**

Passwords maintained by the TSF can be composed using any combination of upper case and lower case letters, numbers, and special characters including the following: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")". The password policy is configurable by a Security Administrator and supports a minimum password length between 8 and 64 characters.

8.3.3 **FIA_UAU_EXT.2:**

Users can authenticate to the TOE locally through the local CLI or remotely through either the remote CLI or the Web GUI.

Local users log in to the local CLI using a username and password through the Serial Port. The credentials are verified using the TOE's local authentication mechanism. The successful verification of the credentials presented to the TOE via the local CLI will provide the user access to all role-based functionality that is assigned to them for the CLI.

Remote users can access the TOE through the remote CLI using username and password or SSH public key. When username and password credentials are used to access the remote CLI, the credentials are verified using the TOE's local authentication mechanism. When public key authentication is used, the TOE authenticates users by verifying the message the TOE receives from the SSH client using the

message's associated public key stored on the TOE. A successful verification of the credentials or public-key presented to the TOE via SSH will provide the user access to all role-based functionality that is assigned to them for the CLI.

Remote users can also access the TOE through the Web GUI using a username and password. The successful verification of the credentials presented to the TOE via HTTPS will provide the user access to all role-based functionality that is assigned to them for the Web GUI.

8.3.4 **FIA_UAU.7:**

While authenticating locally to the TOE, the user's password does not appear in the password field. No echo is present while a user enters the password thus masking the password and preventing the password from being shared. In the case that a user enters invalid credentials (valid/invalid username or valid/invalid password), the TOE does not reveal any information about the invalid component of the credential.

8.3.5 **FIA_UIA_EXT.1:**

In the evaluated configuration, the warning banner is displayed prior to the user authenticating to the TOE via the local CLI, the remote CLI, or the Web GUI. The display of the warning banner is the only service that can be run prior to authentication and thus, the TOE does not allow a user to perform any other actions prior to authentication, regardless of the interface used.

Access is only granted once the user provides valid credentials that are verified using the associated credential authentication mechanism stated in FIA_UAU_EXT.2.

8.3.6 **FIA_X509_EXT.1/Rev, FIA_X509_EXT.2, FIA_X509_EXT.3:**

The TOE performs certificate validity checking for outbound TLS connections to the external audit server and GigaVUE appliances (HTTPS utilizing TLS). A Security Administrator must import the root certificate for each remote TLS server for certificate validation. In addition to the validity checking that is performed by the TOE, the TSF will validate certificate revocation status using the certificate revocation list (CRL) distribution point that is defined in the presented X509 certificate. The TSF will automatically download the CRL from the defined Certification Authority in the Operational Environment, and perform the revocation status check for each of the certificates in the certificate chain. There is no difference in how revocation checking is handled when the TOE is presented with either a full certificate chain or a leaf certificate. In the event that the revocation status cannot be verified, the certificate will not be accepted.

The TSF determines the validity of certificates by ensuring that the certificate and the certificate path are valid in accordance with RFC 5280. In addition, the certificate path is terminated in a trusted CA certificate, the basicConstraints extension is present, and the CA flag is set to TRUE for all CA certificates. The TSF also ensures that the extendedKeyUsage field includes the correct purpose for its intended use. This includes Server Authentication for TLS server certificates; the TSF does not handle certificates associated with TLS clients, OCSP responses, nor code signing certificates.

The Admin role can generate a Certificate Request as specified in RFC 2986 containing the public key and "Common Name" in order for the TOE to have its own TLS Server certificate. The chain of certificates is validated from the root CA when the CA Certificate Response is received.

8.4 Security Management

8.4.1 **FMT_MOF.1/ManualUpdate:**

Software updates are loaded onto the TOE and applied manually. The Admin role is the only administrative role that can perform this action. The TSF restricts the access to this function by enforcing the product's role-based access control system.

8.4.2 **FMT_MTD.1/CoreData:**

The role of Security Administrator is fulfilled by the following roles:

- Admin role for the local and remote CLI interfaces
- Super Admin role for the Web GUI interface

The TSF uses role-based access control to assign each user account to one or more roles, each of which has a fixed set of privileges to interact with the product. Of these roles, only the Admin and Super Admin roles are authorized to perform the management functions associated with the TSF and both roles are functionally identical to the 'Security Administrator' as defined by the NDcPP. The handling of X.509 certificates and managing the certificate trust store are restricted to the Admin and Super Admin through the TSF's role-based access control.

The only security-relevant TOE functionality that is available to a user prior to authentication is the display of the warning banner.

8.4.3 **FMT_MTD.1/CryptoKeys:**

Only Security Administrators are permitted to manipulate cryptographic data on the TOE through the CLI and Web GUI. This behavior is limited to the generating/import and deleting of X.509 certificates and SSH keys as described in Section 8.4.4.

8.4.4 **FMT_SMF.1:**

A user with Admin role is capable of performing management functions on the TOE via the Local CLI and the Remote CLI. The Super Admin role is capable of performing management functions on the TOE via the Web GUI. The following table lists the TSF management functions and identifies the interface(s) that can be used to perform them:

Management Function	Local CLI	Remote CLI	Web GUI
Ability to configure the access banner	Admin vi /etc/issue	Admin vi /etc/issue.net	Super Admin "Settings" → "Preferences" →General Login infobox
Ability to configure the session inactivity time before session termination or locking	Admin vi /etc/ssh/sshd_config	Admin vi /etc/ssh/sshd_config	Super Admin "Settings" → "Preferences" → "Display & Session" Auto logout

Ability to update the TOE, and to verify the updates using hash comparison capability prior to installing those updates	Admin "fmctl image"	Admin "fmctl image"	N/A
Ability to configure the authentication failure parameters for FIA_AFL.1	Admin vi /etc/pam.d/sshd	Admin vi /etc/pam.d/sshd	Super Admin "GigaVUE-FM User Management" → "Authentication" → "Maximum Failed Login Attempts"
Ability to manage the cryptographic keys	Admin CertInstall30.sh run_genEC-key_csr30.sh sudo cp ca.pem sudo cp rservtls-cert.pem sudo cp rservtls-key.pem vi /home/admin/.ssh /authorized_keys ssh-keygen -f /etc/ssh/ssh_host_ecdsa_gigamon_key -t ecdsa -b 521	Admin CertInstall30.sh run_genEC-key_csr30.sh sudo cp ca.pem sudo cp rservtls-cert.pem sudo cp rservtls-key.pem vi /home/admin/.ssh /authorized_keys ssh-keygen -f /etc/ssh/ssh_host_ecdsa_gigamon_key -t ecdsa -b 521	Super Admin "Settings" → "Preferences" → "certificates" → Add/Delete/import (GigaVUE root CA)
Ability to configure thresholds for SSH rekeying	Admin vi /etc /ssh/sshd_config	Admin vi /etc /ssh/sshd_config	N/A
Ability to set the time which is used for time-stamps	Admin date -s "<Day/ Month/ Year> <Hour:Minute:Second>"	Admin date -s "<Day/ Month/ Year> <Hour:Minute:Second>"	N/A
Ability to re-enable an Administrator account	N/A	N/A	Super Admin "Authentication" → "GigaVUE-FM User Management" → "Users"
Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors	Admin CertInstall30.sh run_genEC-key_csr30.sh sudo cp ca.pem sudo cp rservtls-cert.pem sudo cp rservtls-key.pem	Admin CertInstall30.sh run_genEC-key_csr30.sh sudo cp ca.pem sudo cp rservtls-cert.pem sudo cp rservtls-key.pem	Super Admin "Settings" → "Preferences" → "certificates" → Add/Delete/import (GigaVUE root CA)
Ability to import X.509v3 certificates to the TOE's trust store	Admin CertInstall30.sh sudo cp ca.pem sudo cp rservtls-cert.pem sudo cp rservtls-key.pem	Admin CertInstall30.sh sudo cp ca.pem sudo cp rservtls-cert.pem sudo cp rservtls-key.pem	Super Admin "Settings" → "Preferences" → "certificates" → Add/Delete/import (GigaVUE root CA)
Ability to manage the trusted public keys database	Admin vi /home/admin/.ssh	Admin vi /home/admin/.ssh/	N/A

	/authorized_keys ssh-keygen -f /etc/ssh/ssh_host_ecdsa_gigamon_key -t ecdsa -b 521	authorized_keys ssh-keygen -f /etc/ssh/ssh_host_ecdsa_gigamon_key -t ecdsa -b 521	

Table 8-4: Management Functions by Interface

8.4.5 **FMT_SMR.2:**

The security management functions available to authorized users of the TOE are mediated by a role-based access control system. The role-based access control system is enforced through the local CLI, through the remote CLI, and through the Web GUI.

The TOE has two administrative roles: the Admin and the Super Admin. The Admin user is the Security Administrator for the local CLI and remote CLI. The Super Admin is the Security Administrator for the Web GUI. All SFR relevant management activity is performed by these two roles via their respective interfaces. Restrictions on TSF management capabilities for the Super Admin user are described in Table 8-4 above.

8.5 Protection of the TSF

8.5.1 **FPT_APW_EXT.1:**

A hashed representation (SHA-512) of the user password is stored on the TOE. The TOE does not expose plaintext passwords for any user through any interface.

8.5.2 **FPT_SKP_EXT.1:**

The TOE prevents the reading of all pre-shared keys, symmetric keys, and private keys. The TOE does not provide an interface for reading these keys. The TOE's X.509v3 certificate, the key pairs used for SSH communications, and the public certificates used for TLS communication with the audit server and GigaVUE appliances are all stored on the filesystem. These keys are protected by the TOE's role-based access control allowing them to be managed by only the Security Administrator, and direct access to private keys can only be read by the TOE itself. For SSH and TLS sessions, all pre-shared, symmetric keys and private keys are stored in volatile memory and are destroyed once the connection is closed.

8.5.3 **FPT_STM_EXT.1:**

The TOE has a hardware clock that is used for keeping time. A user with the Admin role can set the clock's time manually. The TSF uses time data for the following purposes:

- Audit record timestamps
- Inactivity timeout for administrative sessions
- Expiration checking for certificates
- FIA_AFL.1 timer for lockout duration

8.5.4 **FPT_TST_EXT.1:**

The TOE executes the following series of self-tests automatically during every power on cycle and at the request of a Security Administrator:

BIOS (Basic Input/Output system) power on self-test:

Performs hardware check runs as part of system initialization which includes power-on self-tests of all the major hardware components (e.g., memory, CPU, Ethernet controllers) on the motherboard, including the components that connect to the buses. Failures for any of these checks will result in the platform entering a non-operational state or causing an automatic reboot to attempt to fix and continue startup.

Standard Linux Filesystem Check:

The TOE performs the following checks of the filesystem:

- mounts (creates) basic virtual RAM filesystems
- verifies and mounts the non-volatile filesystem
- verifies and mounts the active or standby software partition filesystem

Failures for any of these checks could result in the platform entering a non-operational state or causing an automatic reboot to attempt to fix and continue startup.

Bouncy Castle Crypto module self-tests:

Cryptographic algorithm testing is provided by Bouncy Castle which enables the TOE's cryptographic functionality. Each time the module is powered up, it performs the pre-operational self-tests to confirm that sensitive data have not been damaged. The pre-operational tests include the Software Integrity test, which verifies the module using HMAC-SHA2-256, and the HMAC and SHS Conditional Cryptographic Algorithm Self-Tests (CAST) which are run prior to the Software Integrity test to ensure the correctness of the HMAC used.

TOE Software integrity:

The software integrity checks are performed through hash verification using SHA-512 and are inclusive of all binaries and libraries that come as part of the product. If any of the components fail the integrity check the TOE audits this information and then the TOE is placed in a non-operational state.

In the event that any of the self-tests fail, the TOE moves into a non-operational state. The user must then contact Gigamon customer support in order bring the TOE back to an operational state.

These tests are sufficient to validate the correct operation of the TSF because they verify that the cryptographic module is operating correctly, the TOE software is valid based on an integrity check prior to use which prevents the software from operating if there is an integrity issue, and that the hardware does not have any anomalies that would cause the software to be executed in an unpredictable or inconsistent manner.

8.5.5 **FPT_TUD_EXT.1:**

The TOE Admin has the ability to query the current executing version and most recently installed version of the TOE's firmware/software on the local CLI and remote CLI. The TOE does not support automatic checking for software updates.

In order to update the TOE, the Admin will need to download the firmware/software update image and identify the associated SHA-256 published hash from an access controlled Gigamon-hosted site. Before

the update image can be loaded on the TOE for installation, the Security Administrator must verify that the published hash of the update image is correct by using a third-party program that supports SHA-256 hashing. If the third party program's hash output value for the firmware/software update image does not match the published hash value, the verification has failed and the update should not be loaded on the TOE for installation and the user should reach to Gigamon for support. If the hash values match, the verification is successful, the Security Administrator can then load and install the update image on the TOE.

The update process can be initiated through either the local CLI or remote CLI. Once the update image has been loaded onto the TOE, the Security Administrator needs to initiate the update process. When the installation process is complete, the TOE does not automatically reboot. The TOE will continue to use the currently executing version of the TOE's software/firmware until a reboot occurs. The Security Administrator must initiate the reboot process to complete the installation of the new update image and have the associated software/firmware become the executing version.

Timely Security Updates

As part of providing timely security updates, Gigamon provides customers with a support section on gigamoncp.force.com/gigmoncp/ where they have the ability to submit support issues. This is an HTTPS site that requires user authentication prior to use. Any feedback that necessitates a fix will result in a patch to Gigamon itself so there is no third-party update process to consider when updating the TOE. Any security fixes will be released as new packages in the same manner as any feature. Any implementation flaws are expected to be addressed within 90 days of reporting. Customers are notified of security-related fixes on the Gigamon customer portal.

8.6 TOE Access

8.6.1 **FTA_SSL_EXT.1:**

The TOE will automatically terminate a local session on the local CLI interface due to inactivity according to a session inactivity timer value set by the Admin. The Admin can configure the local session inactivity timer via the CLI.

A successful automatic session termination of a local user session can be verified through the appearance of a login prompt/notification banner. Once a session has automatically terminated, the user will be required to reauthenticate to the TOE and open a new user session.

8.6.2 **FTA_SSL.3:**

The TOE terminates a remote session for both the remote CLI and Web GUI interfaces due to inactivity according to each interface's respective session inactivity timer configuration. The setting for the remote CLI is configured by the Admin. The setting for the Web GUI is configured by the Super Admin.

A successful automatic session termination of a remote user sessions for both the CLI and Web GUI interfaces can be verified through the appearance of a login prompt/notification banner. Once a session has automatically terminated, the user will be required to reauthenticate to the TOE and open a new user session.

8.6.3 **FTA_SSL.4:**

Any user accessing the TOE is capable of terminating their own session. A user is able to terminate their own session by entering the "exit" command when logged into the local or remote CLI. A user can terminate their own Web GUI session by navigating to the user profile icon and selecting “Logout” from the drop down menu. This will display a confirmation window where the user selects the “Log Out” button.

For all administrative interfaces, a manual user session termination can be verified through the appearance of a login prompt. Once a session has automatically terminated, the user will be required to reauthenticate to the TOE and open a new user session.

8.6.4 **FTA_TAB.1:**

The TOE displays a configurable warning banner on all TOE interfaces prior to authenticating to the TOE. The TOE is accessible through local CLI, remote CLI through SSH, and Web GUI through HTTPS. Each interface’s warning banner must be configured through that particular interface (i.e. Web GUI warning banner must be configured through the Web GUI, etc.).

8.7 Trusted Path/Channels

8.7.1 **FTP_ITC.1:**

The TOE connects and sends data to IT entities that reside in the Operational Environment via trusted channels. In the evaluated configuration, the TOE connects with:

- an external audit server for housing audit records through TLS v1.2
- GigaVUE appliances for the TOE’s primary functionality using HTTPS

In each of these instances, the TOE initiates communication as the client using the cryptographic protocol in the manner described by their respective SFRs. These protocols are used to protect the data traversing the channel from disclosure and/or modification. The remote endpoints are authenticated using TLS server certificates.

8.7.2 **FTP_TRP.1/Admin:**

The Security Administrators are required to authenticate to the TOE in order to be able to perform any management functions. By initiating the trusted path via the remote CLI, Admin users can perform management activities remotely. The remote CLI uses SSHv2 which protects the data traversing the channel from disclosure and/or modification. Super Admin users can also perform management activities remotely via the Web GUI. The Web GUI uses HTTPS which protects the data traversing the channel from disclosure and/or modification.