FireEye CM Series Appliances

FireEye, Inc.
Common Criteria Security Target

Prepared By:
Acumen Security
18504 Office Park Dr
Montgomery Village, MD 20886

www.acumensecurity.net

Table Of Contents

Revision History

Version	Description
1.3	Updated for IAR

1 Security Target Introduction

1.1 Security Target and TOE Reference

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

Category	Identifier
ST Title	FireEye CM Series Appliances Security Target
ST Version	Version 1.3 July 2019
ST Author	Acumen Security, LLC.
TOE Identifier	FireEye CM Series Appliances
TOE Hardware Versions	CM Series Appliances: CM 4500, CM 7500, CM 9500, CM2500V, CM7500V
TOE Software Version	CM Series Appliance Software: 8.0
TOE Developer	FireEye, Inc.
Key Words	Network Device, Security Appliance

Table 1 TOE/ST Identification

1.2 TOE Overview

The FireEye CM Series Appliances (FireEye Email Security) are network devices that secure against advanced email attacks by using signature-less technology to analyze email attachments and quarantine malicious emails.

1.2.1 TOE Product Type

FireEye CM Series Appliances are network devices. Each appliance runs a custom-built hardened version of Linux with only the required services enabled.

1.3 TOE Description

The TOE is comprised of three models of the FireEye CM Series Appliances as shown below.

	CM 4500	CM 7500	CM 9500
Network Ports	2x 10/100/1000BASE-T Ports	2x 10/100/1000BASE-T Ports	2x 10/100/1000BASE-T Ports
Storage	4x 4TB HDD	4x 4TB HDD	8x 4TB HDD
Enclosure	1RU, Fits 19 inch Rack	2RU, Fits 19 inch Rack	2RU, Fits 19 inch Rack

Table 2 CM Series Appliances

	CM2500V	CM7500V
Hypervisor	VMWare ESXi	VMWare ESXi
Platform	Dell 630R	Dell 630R

Table 3 CM Series Virtual Appliances, continued

1.4 TOE Evaluated Configuration

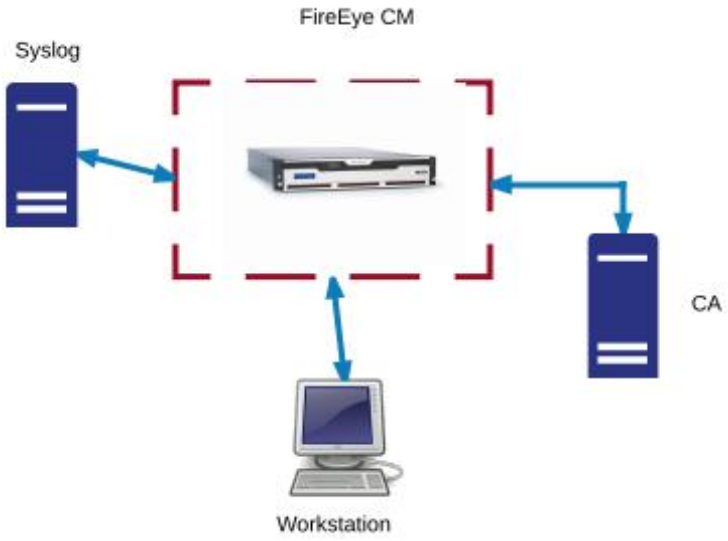
The TOE evaluated configuration consists of one of the appliances listed above. The TOE supports secure connectivity with several IT environment devices as shown in Table 4,

Component	Required	Usage/Purpose Description for TOE performance
Management Workstation with Web Browser/SSH Client	Yes	This includes any IT Environment Management workstation with a Web Browser and a SSH client installed that is used by the TOE administrator to support TOE administration through HTTPS and SSH protected channels. Any SSH client that supports SSHv2 may be used. Any web browser that supports TLS 1.1 or greater may be used.
NTP Server	No	The TOE supports communications with an NTP server to synchronize date and time.

Component	Required	Usage/Purpose Description for TOE performance
Syslog server	No	The syslog audit server is used for remote storage of audit records that have been generated by and transmitted from the TOE. The syslog server must support communications using TLS 1.1 or TLS 1.2.
LDAP AAA Server	No	This includes any IT environment LDAP AAA server that provides authentication services to TOE administrators. The LDAP server must support communications using TLS 1.1 or TLS 1.2.

Table 4 IT Environment Components

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



1.5 TOE Architecture

1.5.1 Physical Boundaries

The TOE is a hardware and software solution that is comprised of the security appliance models described above in section 1.3 The TOE guidance documentation that is considered to be part of the TOE can be found listed in the "FireEye Common Criteria Addendum" document and is posted with the certificate and this Security Target.

The network on which the TOE resides is considered part of the environment. The software is pre-installed and is comprised of only the software versions identified in Section 1.1. In addition, the software images are also downloadable from the FireEye website. A login ID and password is required to download the software image.

1.5.2 Logical Boundaries

The TOE provides the following security functions:

- **Protected Communications.** The TOE protects the integrity and confidentiality of communications as follows:
  - TLS connectivity with the following entities:
    - External LDAP Server (with device level authentication)
    - Audit Server (with device level authentication)
    - Management Web Browser

- SSH connectivity with the following entities:
    - Management SSH Client

- **Secure Administration.** The TOE enables secure local and remote management of its security functions, including:
    - Local console CLI administration
    - Remote CLI administration via SSHv2
    - Remote GUI administration via HTTPS/TLS
    - Administrator authentication using a local database, via LDAP over TLS, or via X.509 certificates to the remote GUI
    - Password complexity enforcement
    - Role Based Access Control - the TOE supports several types of administrative user roles. Collectively these sub-roles comprise the "Security Administrator"
    - Configurable banners to be displayed at login
    - Timeouts to terminate administrative sessions after a set period of inactivity
    - Protection of secret keys and passwords
- **Trusted Update.** The TOE ensures the authenticity and integrity of software updates through digital signatures and requires administrative intervention prior to the software updates being installed.
- **Security Audit.** The TOE keeps local and remote audit records of security relevant events. The TOE internally maintains the date and time which can either be set manually or synchronized with an NTP server. The TOE is configured to transmit its audit messages to an external syslog server. Communication with the syslog server is protected using TLS.
- **Self-Test.** The TOE performs a suite of self-tests to ensure the correct operation and enforcement of its security functions.
- **Cryptographic Operations.** The TOE provides cryptographic support for the services described in Table 5.

Cryptographic Method	Use within the TOE
TLS Establishment	Used to establish initial TLS session.
SSH Establishment	Used to establish initial SSH session.
ECDSA Signature Services	Used in TLS session establishment.
RSA Signature Services	Used in TLS session establishment. Used in SSH session establishment Used in secure software update
SP 800-90 DRBG	Used in TLS session establishment. Used in SSH session establishment
SHS	Used in secure software update
HMAC-SHS	Used to provide TLS traffic integrity verification Used to provide SSH traffic integrity verification
AES	Used to encrypt TLS traffic Used to encrypt SSH traffic

Table 5 TOE Provided Cryptography

Algorithm	CAVP Cert #	Standard	Operation	SFR
RSA	2605	FIPS 186-4	Key Generation	FCS_CKM.1

Algorithm	CAVP Cert #	Standard	Operation	SFR
			Signature Generation/Verification	FCS_COP.1(2)
DSA	1286	FIPS 186-4	Key Generation	FCS_CKM.1
ECDSA	1193	FIPS 186-4	Key Generation Signature Generation/Verification	FCS_CKM.1 FCS_COP.1(2)
SP 800-90 DRBG	1638	SP 800-90A	Random Bit Generation	FCS_RBG_EXT.1
SHS	3904	ISO/IEC 10118-3:2004	Hashing	FCS_COP.1(3)
HMAC-SHS	3172	ISO/IEC 9797-2:2011	Keyed-Hashing	FCS_COP.1(4)
AES	4761	AES specified in ISO 18033-3 CBC specified in ISO 10116 GCM specified in ISO 19772	Encryption/ Decryption	FCS_COP.1(1)
CVL	1406	SP 800-56A	Key Establishment	FCS_CKM.2
RSA	N/A	SP 800-56B (Vendor Affirmed)	Key Establishment	FCS_CKM.2

Table 6 CAVP Algorithm Testing References

Each of the algorithms included in the table above is implemented by the "FireEye Cryptographic Implementation" cryptographic module.

2 Conformance Claims

2.1 CC Conformance

This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 3 conformant

2.2 Protection Profile Conformance

This TOE is conformant to:

- Collaborative Protection Profile for Network Devices, Version 1.0, 27 February 2015 [NDcPP].

2.3 Scheme Interpretations

- 0235 – Applicable because the TOE includes FCS_CKM.2
- 0228 – Applicable because the TOE includes FIA_X509_EXT.1
- 0227 – Applicable because the TOE includes FCS_CKM.1 and acts as a TLS client
- 0226 – Applicable because the TOE includes FCS_TLS*requirements
- 0201 – Applicable because the TOE includes FIA_X509_EXT.1.
- 0199 – Applicable because the TOE includes FCS_COP.1
- 0189 – Applicable because the TOE includes FCS_SSHS_EXT.1
- 0188 – Applicable because the TOE includes FPT_TUD_EXT.1
- 0187 – Applicable because the TOE includes FIA_X509_EXT.1
- 0185 – Applicable because the TOE includes both FPT_TUD_EXT.1 and FTP_ITC.1
- 0184 – Applicable because the TOE includes FIA_X509_EXT.1
- 0183 – Applicable because the NDSD was used for evaluation of the product
- 0182 – Applicable because the TOE includes FIA_X509_EXT.1
- 0181 – Applicable because the TOE includes FPT_TST_EXT.1.
- 0169 – Applicable because the TOE includes FIA_X509_EXT.1
- 0168 – Applicable because the TOE includes FIA_X509_EXT.3
- 0167 – Applicable because the TOE includes FCS_SSHS_EXT.1 and FMT_SMF.1.
- 0165 – Applicable because the TOE includes FCS_TLSC_EXT.1
- 0164 – Applicable because the TOE includes FCS_SSHS_EXT.1
- 0156 – Applicable because the TOE includes FCS_TLSS_EXT.1
- 0155 – Applicable because the TOE includes FCS_TLSS_EXT.1
- 0154 – Applicable because the TOE includes FPT_TUD_EXT.1
- 0153 – Applicable because the TOE includes FPT_STM.1
- 0152 – Applicable because the TOE includes FCS_TLSC_EXT.1
- 0151 – Applicable because the TOE includes FCS_TLSS_EXT.1
- 0150 – Applicable because the TOE includes FCS_SSHS_EXT.1 and FAU_GEN.1
- 0143 – Applicable because the TOE includes FCS_TLSS_EXT.1
- 0130 – Applicable because the TOE includes FCS_CKM.4
- 0126 – Applicable because the TOE includes FCS_TLSC_EXT.1 and FTP_ITC.1
- 0125 – Applicable because the TOE includes FCS_HTTPS_EXT.1

- 0117 – Applicable because the TOE includes FIA_X509_EXT.1
- 0116 – Applicable because the TOE includes FCS_COP.1(2)
- 0114 – Applicable because the TOE includes FCS_COP.1
- 0113 – Applicable because the TOE includes FPT_TUD_EXT.1
- 0112 – Applicable because the TOE includes FCS_TLSS_EXT.1
- 0111 – Applicable because the TOE includes FCS_CKM.1
- 0095 – Applicable because the TOE includes FCS_RBG_EXT.1 FCS_COP.1, and FAU_STG_EXT.1
- 0094 – Applicable because the TOE includes FPT_TUD_EXT.1
- 0090 – Applicable because the TOE includes FPT_TUD_EXT.1 and FMT_SMF.1.

The following technical decisions are not applicable to the evaluation as requirement was not claimed as part of the evaluation:

- 0224 – The TOE does not support IPsec
- 0225 – The TOE does not support IPsec
- 0223 – The TOE does not support IPsec
- 0200 – The TOE does not support SSH client communications
- 0195 – This requirement has been archived/superseded
- 0191 – This requirement has been archived/superseded
- 0186 – The TOE does not support IPsec
- 0170 – The TOE does not support SNMP
- 0160 – The TOE does not support IPsec
- 0115 – The TOE does not support IPsec
- 0096 – The TOE does not include a virtualized instance
- 0093 – This requirement has been archived/superseded

2.4 Conformance Rationale

This Security Target provides exact conformance to the Protection Profile described in the conformance claims above. The security problem definition, security objectives and security requirements in this Security Target are all taken from the applicable Protection Profile performing only operations defined there.

3 Security Problem Definition

The security problem definition has been taken from [NDcPP] and is reproduced here for the convenience of the reader.

3.1 Threats

The threats for the Network Device are grouped according to functional areas of the device in the sections below.

3.1.1 Communications with the Network Device

A network device communicates with other network devices and other network entities. The endpoints of this communication can be geographically and logically distant and may pass through a variety of other systems. The intermediate systems may be untrusted providing an opportunity for unauthorized communication with the network device or for authorized communication to be compromised. The security functionality of the network device must be able to protect any critical network traffic (administration traffic, authentication traffic, audit traffic, etc.). The communication with the network device falls into two categories: authorized communication and unauthorized communication.

Authorized communication includes network traffic allowable by policy destined to and originating from the network device as it was designed and intended. This includes critical network traffic, such as network device administration and communication with an authentication or audit logging server, which requires a secure channel to protect the communication. The security functionality of the network device includes the capability to ensure that only authorized communications are allowed and the capability to provide a secure channel for critical network traffic. Any other communication with the network device is considered unauthorized communication. (Network traffic traversing the network device but not ultimately destined for the device, e.g. packets that are being routed, are not considered to be "communications with the network device" – cf. A.NO_THRU_TRAFFIC_PROTECTION in section 3.2.3.)

The primary threats to network device communications addressed in [the NDcPP] focus on an external, unauthorized entity attempting to access, modify, or otherwise disclose the critical network traffic. A poor choice of cryptographic algorithms or the use of non-standardized tunnelling protocols along with weak Administrator credentials, such as an easily guessable password or use of a default password, will allow a threat agent unauthorized access to the device. Weak or no cryptography provides little to no protection of the traffic allowing a threat agent to read, manipulate and/or control the critical data with little effort. Non-standardized tunnelling protocols not only limit the interoperability of the device but lack the assurance and confidence standardization provides through peer review.

3.1.1.1 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.

3.1.1.2 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.

3.1.1.3 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.

3.1.1.4 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.

3.1.2 Valid Updates

Updating network device software and firmware is necessary to ensure that the security functionality of the network device is maintained. The source and content of an update to be applied must be validated by cryptographic means; otherwise, an invalid source can write their own firmware or software updates that circumvents the security functionality of the network device. Methods of validating the source and content of a software or firmware update by cryptographic means typically involve cryptographic signature schemes where hashes of the updates are digitally signed.

Unpatched versions of software or firmware leave the network device susceptible to threat agents attempting to circumvent the security functionality using known vulnerabilities. Nonvalidated updates or updates validated using non-secure or weak cryptography leave the updated software or firmware vulnerable to threat agents attempting to modify the software or firmware to their advantage.

3.1.2.1 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.

3.1.3 Audited Activity

Auditing of network device activities is a valuable tool for Administrators to monitor the status of the device. It provides the means for Administrator accountability, security functionality activity reporting, reconstruction of events, and problem analysis. Processing performed in response to device activities may give indications of a failure or compromise of the security functionality. When indications of activity that impact the security functionality are not generated and monitored, it is possible for such activities to occur without Administrator awareness. Further, if records are not generated and retained, reconstruction of the network and the ability to understand the extent of any compromise could be negatively affected.

Additional concerns are the protection of the audit data that is recorded from alteration or unauthorized deletion. This could occur within the TOE, or while the audit data is in transit to an external storage device.

Note [the NDcPP] requires that the network device generate the audit data and have the capability to send the audit data to a trusted network entity (e.g., a syslog server).

3.1.3.1 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.

3.1.4 Administrator and Device Credentials Data

A network device contains data and credentials which must be securely stored and must appropriately restrict access to authorized entities. Examples include the device firmware, software, configuration authentication credentials for secure channels, and Administrator credentials. Device and Administrator keys, key material, and authentication credentials need to be protected from unauthorized disclosure and modification. Furthermore, the security functionality of the device needs to require default authentication credentials, such as Administrator passwords, be changed.

Lack of secure storage and improper handling of credentials and data, such as unencrypted credentials inside configuration files or access to secure channel session keys, can allow an attacker to not only gain access to the network device, but also compromise the security of the network through seemingly authorized modifications to configuration or though man-in-the middle attacks. These attacks allow an unauthorized entity to gain access and perform administrative functions using the Security Administrator's credentials and to intercept all traffic as an authorized endpoint. This results in difficulty in detection of security compromise and in reconstruction of the network, potentially allowing continued unauthorized access to Administrator and device data.

3.1.4.1 T.SECURITY_FUNCTIONALITY_COMPROMISE

Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.

3.1.4.2 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.

3.1.5 Device Failure

Security mechanisms of the network device generally build up from roots of trust to more complex sets of mechanisms. Failures could result in a compromise to the security functionality of the device. A network device self-testing its security critical components at both start-up and during run-time ensures the reliability of the device's security functionality.

3.1.5.1 T.SECURITY_FUNCTIONALITY_FAILURE

A component of the network device may fail during start-up or during operations causing a compromise or failure in the security functionality of the network device, leaving the device susceptible to attackers.

3.2 Assumptions

This section describes the assumptions made in identification of the threats and security requirements for network devices. The network device is not expected to provide assurance in any of these areas, and as a result, requirements are not included to mitigate the threats associated.

3.2.1 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 and/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 [NDcPP] will not include any requirements on physical tamper protection or other physical attack mitigations. The [NDcPP] will 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.

3.2.2 A.LIMITED_FUNCTIONALITY

The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).

3.2.3 A.NO_THRU_TRAFFIC_PROTECTION

A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the NDcPP. It is assumed that this protection will be covered by cPPs for particular types of network devices (e.g., firewall).

3.2.4 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 being 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.

3.2.5 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.

3.2.6 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.

3.3 Organizational Security Policy

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. The description of each policy is described in the section below.

3.3.1 P.ACCESS_BANNER

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

4 Security Objectives

The security objectives have been taken from [NDcPP] and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the Operational Environment

The following subsections describe objectives for the Operational Environment.

4.1.1 OE.PHYSICAL

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

4.1.2 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.

4.1.3 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.

4.1.4 OE.TRUSTED_ADMIN

Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

4.1.5 OE.UPDATES

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

4.1.6 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.

5 Security Requirements

This section identifies the Security Functional Requirements for the TOE and/or Platform. The Security Functional Requirements included in this section are derived from the CC Part 2 and all applicable Protection Profiles as described in section 2.

5.1 Conventions

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

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

Explicitly stated SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs. Formatting conventions outside of operations matches the formatting specified within the PP.

5.2 TOE Security Functional Requirements

This section identifies the Security Functional Requirements for the TOE. The TOE Security Functional Requirements that appear below in Table 7 are described in more detail in the following subsections.

5.2.1 Class: Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) *All administrative actions comprising:*
    - *Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).*
    - *Security related 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).*
    - *Starting and stopping services (if applicable)*
    - *[*no other actions*];*
d) *Specifically defined auditable events listed in Table 7.*

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, *information specified in column three of Table 7*.

Requirement	Auditable Events	Additional Audit Record Contents
Mandatory SFRs		
FAU_GEN.1	None.	None.
FAU_GEN.2	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(1)	None.	None.
FCS_COP.1(2)	None.	None.
FCS_COP.1(3)	None.	None.
FCS_COP.1(4)	None.	None.
FCS_RBG_EXT.1	None.	None.
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None
FIA_X509_EXT.1	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1(1)/Trusted Update	Any attempt to initiate a manual update	None.
FMT_MTD.1	All management activities of TSF data.	None.
FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM.1	Changes to time.	The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.
Selection-Based SFRs		
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
FCS_TLSS_EXT.2	Failure to establish a TLS Session	Reason for failure

Table 7 TOE Security Functional Requirements and Auditable Events

FAU_GEN.2 User Identity Association

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

FAU_STG_EXT.1 External Audit Trail Storage

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

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

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

5.2.2 Class: Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

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

- **RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following:**

FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;

- **ECC schemes using "NIST curves" [P-256] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;**
- **FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1**

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

FCS_CKM.2 Cryptographic Key Establishment

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

- **RSA-based key establishment schemes that meet the following: NIST Special Publication 800-56B Revision 1, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography";**
- **Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";**
- **Finite field-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";**

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

FCS_CKM.4 Cryptographic Key Destruction

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

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

that meets the following: *No Standard*.

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

FCS_COP.1.1(1) The TSF shall perform **encryption/decryption** in accordance with a specified cryptographic algorithm [*AES used in [CBC, GCM] mode]* and cryptographic key sizes [*128 bits, 256 bits*] that meet the following: [*AES as specified in ISO 18033-3, [CBC as specified in ISO 10116, GCM as*

specified in ISO 19772]].

FCS_COP.1(2) Cryptographic Operation (Signature Generation and Verification)

FCS_COP.1.1(2) The TSF shall perform **cryptographic signature services (generation and verification)** in accordance with a specified cryptographic algorithm [

- RSA Digital Signature Algorithm and cryptographic key sizes (**modulus**) [*2048 bits or greater*],
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes [*256 bits or greater*]

]

that meet the following: [

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

].

FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)

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

FCS_COP.1(4) Cryptographic Operation (Keyed-Hash Algorithm)

FCS_COP.1.1(4) The TSF shall perform **keyed-hash message authentication** in accordance with a specified cryptographic algorithm [HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512] and cryptographic key sizes [*512 bits, 1024 bits*] **and message digest sizes [160, 256, 384, 512] bits** that meet the following: [*ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*].

FCS_HTTPS_EXT.1 HTTPS Protocol

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

FCS_HTTPS_EXT.1.2 The application shall implement HTTPS using TLS in accordance with [FCS_TLSS_EXT.1].

FCS_HTTPS_EXT.1.3 The TSF shall establish the connection only if [the peer initiates handshake].

Application Note: *For TSF HTTPS clients, no mutual authentication is required or enforced. For TSF servers – specifically, the remote web GUI – the server, if configured, can require an X.509 certificate from the client as per FCS_TLSS_EXT.2.5 below. Irrespective, any TSF client or server that encounters*

an invalid peer certificate shall not establish the connection.

FCS_RBG_EXT.1 Random Bit Generation

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

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[*four*] software-based noise source**s**] with a minimum of [128 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions", of the keys and hashes that it will generate.

FCS_SSHS_EXT.1 SSH Server Protocol

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

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

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*65,536*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [aes128-cbc, aes256-cbc, AEAD_AES_128_GCM, AEAD_AES_256_GCM].

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [ssh-rsa] and [no other public key algorithms] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [hmac-sha1] and [hmac-sha2-256, hmac-sha2-512, AEAD_AES_128_GCM, AEAD_AES_256_GCM] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [diffie-hellman-group14-sha1] 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 no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

FCS_TLSC_EXT.1 TLS Client Protocol

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

- *Mandatory Ciphersuites:*
    - *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*

[

- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

].

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125.

FCS_TLSC_EXT.1.3 The TSF shall only establish a trusted channel if the server certificate is valid.

FCS_TLSC_EXT.1.4
The TSF shall present the Supported Elliptic Curves Extension in the
Client Hello with the following NIST curves: ***[secp256r1]*** and no other curves

FCS_TLSS_EXT.1 TLS Server Protocol

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

- *Mandatory Ciphersuites:*
- *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
  [
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

].

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

[none].

FCS_TLSS_EXT.1.3 The TSF shall generate key establishment parameters using RSA with key size 2048 bits and [3072-bits].

***Application Note**: This instance of FCS_TLSS_EXT.1 is applicable when the remote web GUI is configured without support for client-side X.509 authentication. If configured with client-side X.509 authentication support, FCS_TLSS_EXT.2 shall be enforced.*

FCS_TLSS_EXT.2 TLS Server Protocol with mutual authentication

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

- *Mandatory Ciphersuites:*
  - TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

[

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

].

FCS_TLSS_EXT.2.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [none].

FCS_TLSS_EXT.2.3 The TSF shall generate key establishment parameters using RSA with key size 2048 bits and [3072 bits] and [over NIST curves [secp256r1, secp384r1] and no other curves; Diffie-Hellman parameters of size 2048 bits and [3072 bits]].

FCS_TLSS_EXT.2.4 The TSF shall support mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.2.5 The TSF shall not establish a trusted channel if the client certificate is invalid.

FCS_TLSS_EXT.2.6 The TSF shall not establish a trusted channel if the distinguished name (DN) or Subject Alternative Name (SAN) contained in a certificate does not match the expected identifier for the peer.

5.2.3 Class: Identification and Authentication (FIA)

FIA_PMG_EXT.1 Password Management

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

1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, *[“”[1], “+”, “-“, “.”, “/”, “:”, “;”, “<”, “=”, “>”, “?”, “\”, “[“, “]”[2], “^”, “_”[3], “`”[4], “{“, “|”[5], “}”, and “~”*]];
2. *Minimum password length shall settable by the Security Administrator, and shall support passwords of 15 characters or greater;*

FIA_UIA_EXT.1 User Identification and Authentication

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

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

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

FIA_UAU_EXT.2 Password-based Authentication Mechanism

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

FIA_UAU.7 Protected Authentication Feedback

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

FIA_X509_EXT.1 Certificate Validation

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

- RFC 5280 certificate validation and certificate path.
- The certificate path must terminate with a trusted CA certificate.

[1] Single-quote character
[2] Left and right square brackets (the bottom part of the square bracket hidden by the underlying convention of the selection operation).
[3] Underscore, which is hidden by the underlining convention of the selection operation.
[4] Backtick character
[5] Vertical bar/pipe character

- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [a 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 The TSF shall only treat a certificate as a CA certificate if the basicConstraints extension is present and the CA flag is set to TRUE.

FIA_X509_EXT.2 X.509 Certificate Authentication

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

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

FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3.1 The TSF shall generate a Certificate Request Message 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.

5.2.4 Class: Security Management (FMT)

FMT_MOF.1(1)/TrustedUpdate Management of security functions behavior

FMT_MOF.1.1(1)/TrustedUpdate The TSF shall restrict the ability to **enable** the functions **to perform manual updates** to **Security Administrators**.

FMT_MTD.1 Management of TSF Data

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

FMT_SMF.1 Specification of Management Functions

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

[

- Ability to administer the TOE locally and remotely;
- Ability to configure the access banner;
- Ability to configure the session inactivity time before session termination or locking;
- Ability to update the TOE, and to verify the updates using [digital Signature] capability prior to installing those updates;
- [
    - *Ability to configure the cryptographic functionality;*]

].

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 the user with roles.

FMT_SMR.2.3 The TSF shall ensure that the conditions

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

are satisfied.

5.2.5 Class: Protection of the TSF (FPT)

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

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

FPT_APW_EXT.1 Protection of Administrator Passwords

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

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

FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [during initial start-up (on power on)] to demonstrate the correct operation of the TSF: [*POST*, *Cryptographic Tests, Software Integrity Test*].

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 [digital signature mechanism] prior to installing those updates.

FPT_STM.1 Reliable Time Stamps

FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

5.2.6 Class: TOE Access (FTA)

FTA_SSL_EXT.1 TSF-initiated Session Locking

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

- terminate the session]

after a Security Administrator-specified time period of inactivity.

FTA_SSL.3 TSF-initiated Termination

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

FTA_SSL.4 User-initiated Termination

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

FTA_TAB.1 Default TOE Access Banners

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

5.2.7 Class: Trusted Path/Channels (FTP)

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall be **capable of using [TLS] to provide** a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [authentication server]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [*audit server, authentication server*].

FTP_TRP.1 Trusted Path

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

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

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

5.3 TOE SFR Dependencies Rationale for SFRs

The Collaborative Protection Profile for Network Devices contains all the requirements claimed in this Security Target. As such, SFR dependencies are not applicable since the PP has been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the Collaborative Protection Profile for Network Devices. The assurance requirements are summarized in the table below.

Assurance Class	Components	Components Description
Security Target	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Components Definition
	ASE_INT.1	ST Introduction
	ASE_OBJ.1	Security Objectives for the Operational Environment

	ASE_REQ.1	Stated Security Requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Life Cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests	ATE_IND.1	Independent Testing – Conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

Table 8 Security Assurance Requirements

5.5 Rationale for Security Assurance Requirements

The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.

5.6 Assurance Measures

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

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

SAR Component	How the SAR will be met
	how potential changes are incorporated, and the degree to which automation is used to reduce the scope for error.
ATE_IND.1	FireEye will provide the TOE for testing.
AVA_VAN.1	FireEye will provide the TOE for testing.

Table 9 TOE Security Assurance Measures

6 TOE Summary Specification

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

TOE SFR	Rationale
FAU_GEN.1	The TOE generates a comprehensive set of audit logs that identify specific TOE operations whenever an auditable event occurs. Auditable events are specified at Table 7. Each of the events is specified in the audit record is in enough detail to identify the user for which the event is associated, when the event occurred, where the event occurred, the outcome of the event, and the type of event that occurred. The audit trail consists of the individual audit records; one audit record for each event that occurred. As noted above, the information includes [at least] all of the required information. The log buffer is circular, so newer messages overwrite older messages after the buffer is full. Administrators are instructed to monitor the log buffer to view the audit records. The first message displayed is the oldest message in the buffer. The TOE does not have an interface to modify audit records, though there is an interface available for the authorized administrator to clear audit data stored locally on the TOE.
FAU_GEN.2	The TOE ensures that each auditable event is associated with the user that triggered the event. For example, a human user, user identity or related session ID would be included in the audit record. For an IT entity or device, the IP address, MAC address, host name, or other configured identification is included in the audit record.
FAU_STG_EXT.1	The TOE may be configured to export syslog records to a specified, external syslog server. The TOE also stores a limited set of audit records locally on the TOE, and continues to do so if the communication with the syslog server goes down. The TOE protects communications with an external syslog server via TLS. The TOE transmits its audit events to all configured syslog servers at the same time logs are written locally to non-volatile storage. If the TLS connection fails, the TOE continues to store audit records locally on the TOE, and will transmit any locally stored contents when connectivity to the syslog server is restored. Only Authorized Administrators are able to clear the local logs, and local audit records are stored in a directory that does not allow administrators to modify the contents. The amount of audit data that can be stored locally is configurable by setting the local log rotation parameters (e.g. see the `logging files rotation` CLI commands). When the local log is full, the oldest log files are deleted to allow a new log to be created.
FCS_CKM.1	In support of secure cryptographic protocols, the TOE supports RSA key generation schemes as specified in NIST SP-800-186-4, with key sizes of 2048 and 3072 bits. These keys are used in support of digital certificates for both TLS and SSH.

	Additionally, the TOE supports Elliptic Curve key generation of p-256. The keys are used in support of ECDH key exchange as part of TLS. The relevant NIST CAVP certificate numbers are listed in Table 6. The vendor claims NIST SP 800-56B conformance via vendor affirmation.
FCS_CKM.2	In support of secure cryptographic protocols, the TOE supports several key establishment schemes, including, • RSA based key exchange based on NIST SP 800-56Br1; • ECC based key exchange based on NIST SP 800-56Ar2; • FFC based key exchange based on NIST SP 800-56Ar2 RSA, ECC and FFC schemes are used for TLS. The TSF claims Diffie Hellman group 14 for SSH key exchange which is implemented by hardcoding Oakley Group 14 parameters as defined in RFC3526, section 3. The TOE acts as a sender and receiver for all schemes. The relevant NIST CAVP certificate numbers are listed in Table 6.
FCS_CKM.4	The TOE meets all requirements specified in NDcPP for destruction of keys and Critical Security Parameters (CSPs). All keys within the TOE are securely destroyed as per the descriptions given in Table 11 below.
FCS_COP.1(1)	The TOE provides symmetric encryption and decryption capabilities using 128 and 256 bit AES in CBC mode, and GCM mode as described in NIST SP 800-38A and NIST SP 800-38D, respectively. AES is implemented in the following protocols: TLS and SSH. The relevant NIST CAVP certificate numbers are listed in Table 6.
FCS_COP.1(2)	The TOE provides cryptographic signature generation and verification services using • RSA Signature Algorithm with key size of 2048 and greater, • ECDSA Signature Algorithm with NIST curves P-256, P-384 and P-521. These RSA and ECDSA signature verification services are used in the TLS protocols. Additionally, RSA signature verification is used for the SSH protocol (ssh-rsa). The relevant NIST CAVP certificate numbers are listed in Table 6.
FCS_COP.1(3)	The TOE provides cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512 as specified in FIPS Pub 180-4 “Secure Hash Standard.” SHS is implemented in the following parts of the TSF: • TLS and SSH; • Digital signature verification as part of trusted update validation; and • Hashing of passwords in non-volatile storage. The relevant NIST CAVP certificate numbers are listed in Table 6.
FCS_COP.1(4)	The TOE provides keyed-hashing message authentication services using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 as specified in FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code,” and FIPS 180-4, “Secure Hash Standard.”

<table>
<tr><td></td><td>HMAC is implemented in the following protocols: TLS and SSH.

The characteristics of the HMACs used in the TOE are given in the following table:

<table>
<tr><th>Algorithm</th><th>Hash function</th><th>Block size</th><th>Key size</th><th>Digest size</th></tr>
<tr><td>HMAC-SHA-1</td><td>SHA-1</td><td>512 bits</td><td>512 bits</td><td>160 bits</td></tr>
<tr><td>HMAC-SHA-256</td><td>SHA-256</td><td>512 bits</td><td>512 bits</td><td>256 bits</td></tr>
<tr><td>HMAC-SHA-384</td><td>SHA-384</td><td>1024 bits</td><td>1024 bits</td><td>384 bits</td></tr>
<tr><td>HMAC-SHA-512</td><td>SHA-512</td><td>1024 bits</td><td>1024 bits</td><td>512 bits</td></tr>
</table>

The relevant NIST CAVP certificate numbers are listed in Table 6.</td></tr>
<tr><td>FCS_HTTPS_EXT.1</td><td>The TOE provides management functionality over an HTTPS connection using the TLS implementation described above. The TOE is therefore subject to claiming FCS_HTTPS_EXT.1 in a server capacity. The TOE does not use HTTPS in a client capacity. The TOE's HTTPS protocol complies with RFC 2818.

RFC 2818 is, quite simply, HTTP over TLS. The majority of RFC 2818 is spent on discussing practices for validating endpoint identities and how connections must be setup and torn down. The TOE web GUI operates on an explicit port designed to natively speak TLS: it does not attempt STARTTLS or similar multi-protocol negotiation which is described in section 2.3 of RFC 2818. The web server uses a variant of OpenSSL which attempts to send closure Alerts prior to closing a connection in accordance with section 2.2.2 of RFC 2818.[6] Finally, client identification is performed via username and password as described in FIA_UIA_EXT.2</td></tr>
<tr><td>FCS_RBG_EXT.1</td><td>The TOE implements a NIST-approved AES-CTR Deterministic Random Bit Generator (DRBG), as specified in SP 800-90.

The entropy source used to seed the Deterministic Random Bit Generator is a random set of bits regularly supplied to the DRBG from three software sources. (This ST considers the sources 'software' simply because the entropy sources are not considered True Random Number Generators (TRNGs) based on random properties of physical processes.) The combined 256-bit seed value contains 256 bits of independent and identically distributed (IID) entropy.

All RNG entropy source samplings are continuously health tested by the NIST DRBG as per SP 900-90A before using them as a seed.

The relevant NIST CAVP certificate numbers are listed in Table 6.</td></tr>
<tr><td>FCS_SSHS_EXT.1</td><td>The TOE uses SSH in a server capacity to remotely manage by an administrator-controlled (out-of-scope) SSH client.

The SSH server is capable of using both RSA public keys (ssh-rsa) and passwords for client authentication to the remote server. Public keys can be associated to named administrative users.

Large SSH packets are defined as those greater than 65,535 bytes. This is accomplished by buffering all data for a particular SSH packet transmission until the buffer limit is reached and then dropping the packet if this limit is exceeded.</td></tr>
</table>

[6] https://www.openssl.org/docs/standards.html

	The TOE supports the following cryptographic algorithms: • AES-CBC-128, AES-CBC-256, AEAD_AES_128_GCM, and AEAD_AES_256_GCM to ensure confidentiality of the session; • HMAC-SHA1, HMAC-SHA2-256, HMAC-SHA2-512, AEAD_AES_128_GCM and AEAD_AES_256_GCM; Keys are exchanged between the client and server using diffie-hellman-group14-sha1. The TOE SSH server is capable of rekeying. The TOE implements two thresholds: • When 1 GB of aggregate data is transferred between the client-server pair (irrespective of direction of data flow); and • When 1 hour has elapsed. The TOE continuously checks both conditions. When either of the conditions are met, the TOE will initiate a rekey. All session keys are rekeyed at the same time (eg. confidentiality and integrity keys). The TOE server maintains an SSH server hostkey fingerprint which can be used by an SSH client to detect server authenticity.
FCS_TLSC_EXT.1	The TOE has two trusted channels which make use of TLS: • LDAP; and • Syslog. Both the LDAP and the syslog channel clients allow TLS protocol versions 1.1 and 1.2 and are both restricted to the following ciphersuites by default: • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_DHE_RSA_WITH_AES_128_CBC_SHA • TLS_DHE_RSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256 • TLS_RSA_WITH_AES_256_CBC_ SHA256 • TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 • TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 Ciphersuites are not user-configurable. The reference identifier for both the external LDAP server and syslog server are configured by the administrator using the available administrative commands in the CLI or the web GUI. When either the LDAP client or syslog client receives an X.509 certificate from their respective servers, the client will compare the reference identifier with the established Subject Alternative Names (SANs) in the

	certificate. If a SAN is available and does not match the reference identifier, then the verification fails and the channel is terminated. If there are no SANs of the correct type in the certificate, then the TSF will compare the reference identifier to the Common Name (CN) in the certificate Subject. If there is no CN, then the verification fails and the channel is terminated. If the CN exists and does not match, then the verification fails and the channel is terminated. Otherwise, the reference identifier verification passes and additional verification actions can proceed. Neither the LDAP nor syslog TLS clients support certificate pinning. Both LDAP and syslog TLS clients will transmit the Supported Elliptic Curves extension in the Client Hello message by default with support for the following NIST curves: secp256r1. The non-TOE server can choose to negotiate the elliptic curve from this set for any of the mutually negotiable elliptic curve ciphersuites.
FCS_TLSS_EXT.1 FCS_TLSS_EXT.2	The TOE has a single trusted path over the remote web GUI which acts as a TLS server. This server can be optionally configured to allow authentication using X.509 certificates. The server only allows TLS protocol versions 1.1 and 1.2 (rejecting any other protocol version, including SSL 2.0, SSL 3.0 and TLS 1.0 and any other unknown TLS version string supplied) and is restricted to the following ciphersuites by default: • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_DHE_RSA_WITH_AES_128_CBC_SHA • TLS_DHE_RSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256 • TLS_RSA_WITH_AES_256_CBC_ SHA256 • TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256 • TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 Ciphersuites are not user-configurable. The TLS server is capable of negotiating ciphersuites that include RSA, DHE, and ECDHE key agreement schemes. The RSA key agreement parameters are provided by the associated RSA certificate loaded to the server. The server certificate is restricted to being 2048 bits or 3072 bits. The DHE key agreement parameters are restricted to 2048 bits and are hardcoded into the server. When a non-TOE TLS client initiates communication with the TOE TLS server

	configured for mutual authentication, the server requires a certificate be sent from the client in the Client Certificate handshaking message. That client certificate must have at least one of the following entities contained: • A Common Name (CN) in the X.509 Subject composed as an arbitrary username string; • An email as a Subject Alternative Name (SAN) email type composed as user@example.com; • A UPN encoded as a SAN 'other name'/msUPN type (OID 1.3.6.1.4.1.311.20.2.3) composed as user@example.com The TOE configured for mutual authentication ensures that the associated user accounts have a mapping describing which X.509 entity will be compared for that user. The TOE has the ability to extract the username portion from any RFC 822-formatted names[7]. By default, the UPN SAN is used. When the server configured for mutual authentication receives an X.509 certificate from the client, certificate will be parsed to determine if the appropriate field exists in the certificate. If a SAN is available and does not match the reference identifier, then the verification fails and the channel is terminated. If there are no appropriate SANs in the certificate, then the TSF will compare the reference identifier to the Common Name (CN) in the certificate Subject. If there is no CN, then the verification fails and the channel is terminated. If the CN exists and does not match, then the verification fails and the channel is terminated. Otherwise, the reference identifier verification passes and additional verification actions can proceed.
FIA_PMG_EXT.1	The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", "'", "+", "-", ".", "/", ":", ";", "<", "=", ">", "?", "[", "\", "]", "^", "_", "`", "{", "|", "}", and "~". The minimum password length is settable by the Authorized Administrator and can range from 8 to 32 characters.
FIA_UIA_EXT.1 FIA_UAU_EXT.2	The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed. Administrative access to the TOE is facilitated through one of several interfaces, • Directly connecting to each TOE appliance • Remotely connecting to each appliance via SSHv2 • Remotely connecting to appliance GUI via HTTPS/TLS Regardless of the interface at which the administrator interacts, the TOE prompts the user for a credential. Only after the administrative user presents the correct authentication credentials will they be granted access to the TOE administrative functionality. No TOE administrative access is permitted until an administrator is successfully identified and authenticated. The TOE provides a local password based authentication mechanism as well as LDAP authentication, if configured.

	The process for authentication is the same for administrative access whether administration is occurring via direct connection or remotely. For mechanisms other than X.509 mutual authentication to the remote web GUI, at initial login, the administrative user is prompted to provide a username. After the user provides the username, the user is prompted to provide the administrative credential associated with the user account (eg. password or SSH public/private key response). The TOE then either grants administrative access (if the combination of username and credential is correct) or indicates that the login was unsuccessful. The TOE does not provide a reason for failure in the cases of a login failure. For X.509-based client authentication to the web GUI, the non-TOE web client (the 'web browser') is responsible for acquiring the X.509 certificate from the user and transmitting it to the TOE via the TLS client Certificate protocol message. The TOE will validate the client certificate according to the rules for FIA_X509_EXT.1 and ensure the encoded subject is associated with an administrative account on the TOE. If the X.509 certificate fails to validate or the subject is not associated with a permitted administrative account, the TOE will reject the X.509 credential. The TOE does not permit any administrative function to be accessible until after an administrator is successfully identified and authenticated.
FIA_UAU.7	For all authentication at the local CLI the TOE displays only "*" characters when the administrative password is entered so that the password is obscured.
FIA_X509_EXT.1 FIA_X509_EXT.2 FIA_X509_EXT.3	The TOE performs X.509 certificate validation at the following three points: • TOE TLS client authentication of server X.509 certificates; • TOE TLS server authentication of client X.509 certificates; • When certificates are loaded into the TOE, such as when importing CAs, certificate responses and other device-level certificates (such as the web server certificate presented by the TOE TLS web GUI). In all three scenarios, certificates are checked for several validation characteristics: • If the certificate 'notAfter' date is in the past, then this is an expired certificate which is considered invalid; • The certificate chain must terminate with a trusted CA certificate; • Server certificates consumed by the TOE TLS client must have a 'serverAuthentication' extendedKeyUsage purpose; • Client certificates consumed by the TOE TLS server (for mutual authentication) must have a 'clientAuthentication' extendedKeyUsage purpose; A trusted CA certificate is defined as any certificate loaded into the TOE trust store that has, at a minimum, a basicConstraints extension with the CA flag set to TRUE. Certificate revocation checking is performed using Certificate Revocation Lists responders. The CRL must have the cRLsigning bit set.

	As X.509 certificates are not used for either trusted updates or firmware integrity self-tests, the code-signing purpose is not checked for in the extendedKeyUsage. The TOE has a trust store where root CA and intermediate CA certificates can be stored. The trust store is not cached: if a certificate is deleted, it is immediately untrusted. If a certificate is added to the trust store, it is immediately trusted for its given scope. The X.509 certificates for each of the given scenarios are validated using the certificate path validation algorithm defined in RFC 5280, which can be summarized as follows: • The public key algorithm and parameters are checked • The current date/time is checked against the validity period revocation status is checked • Issuer name of X matches the subject name of X+1 • Name constraints are checked • Policy OIDs are checked • Policy constraints are checked; issuers are ensured to have CA signing bits • Path length is checked • Critical extensions are processed If, during the entire trust chain verification activity, any certificate under review fails a verification check, then the entire trust chain is deemed untrusted and the TLS connection is terminated. As part of the verification process, Certificate Revocation Lists are used to determine whether the certificate is revoked or not. Instructions for configuring the trusted IT entities to supply appropriate X.509 certificates are captured in the guidance documents. The TOE is capable of generating certificate signing requests (CSRs). The user can select the size of the key as 2048 or 3072 bits. In addition to adding the public key to the certificate details, the user can provide information for the Common Name, Organization, Organizational Unit, Locality, State/Province, Country, and Email Address. No device-specific details are collected and added to the certificate request to be signed.
FMT_MOF.1(1)/ TrustedUpdate	The TOE restricts the ability to perform software updates to the Admin role.
FMT_MTD.1 FMT_SMR.2	The TOE implements role based access control. Administrative users are required to login before being provided with access to any administrative functions. The TOE supports several types of administrative user roles. Collectively these sub-roles comprise the Security Administrator. The supported roles include, • Admin: The system administrator is a “super user” who has all capabilities. The primary function of this role is to configure the system. • Monitor: The system monitor has read-only access to some things the admin role can change or configure.

	• Operator: The system operator has a subset of the capabilities associated with the admin role. Its primary function is configuring and monitoring the system • Analyst: The system analyst focuses on data plane analysis and possesses several capabilities, including setting up alerts and reports. • Auditor: The system auditor reviews audit logs and performs forensic analysis to trace how events occurred. Each of the predefined administrative sub-roles have a set of permissions that will grant them access to the TOE data, though with some sub-roles, the access is limited. The TOE performs role-based authorization, using TOE platform authorization mechanisms, to grant access to the privileged and semi-privileged levels. The term "Security Administrator" is used in this ST to refer to any user which has been assigned a sub-role that is permitted to perform the relevant action; therefore has the appropriate privileges to perform the requested functions.
FMT_SMF.1	The TOE may be managed via the CLI (console & SSH) or GUI (HTTPS). The specific management capabilities include: • Ability to administer the TOE locally and remotely (GUI & CLI); • Ability to configure the access banner (GUI & CLI); • Ability to configure the session inactivity time before session termination or locking (GUI & CLI); • Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates (CLI & GUI);
FPT_SKP_EXT.1	The TOE stores all private keys in a secure directory that is not readily accessible to administrators; hence no interface access. Refer to section 6.1 for key storage details.
FPT_APW_EXT.1	The TOE stores Security Administrator passwords. All passwords are stored in a secure directory that is not readily accessible to administrators. The passwords are stored SHA-512 hashed and not in plaintext. There are no administrative interfaces available that allow passwords to be viewed.
FPT_TST_EXT.1	The TOE runs a suite of self-tests during initial start-up to verify its correct operation. If any of the tests fail, the TOE will enter into an error state until an Administrator intervenes. During the system bootup process (power on or reboot), all the Power on Startup Test (POST) components for all the cryptographic modules perform the POST. The Software Integrity Test is run automatically on start-up, and whenever the system images are loaded. A hash verification is used to confirm the image file to be loaded has not been corrupted and has maintained its integrity. These tests are sufficient to verify that the correct version of the TOE software is running as well as that the cryptographic operations are all performing as expected. Both of these functions are required to ensure that the TOE is operating as expected and data that the user expects to be encrypted in not transferred in plaintext.

FPT_TUD_EXT.1	The Security Administrator can query the software version running on the TOE and the most recently downloaded software version. When software updates are made available by FireEye the Security Administrator can obtain, verify the integrity of, and install those updates. Software updates are downloaded to the TOE via an `fenet image fetch` command on the CLI or by navigating to the `Update` page in the web GUI. Software images will not be installed without explicit administrative intervention. The TOE image files are digitally signed (2048-bit RSA/SHA-256) so their integrity can be verified during the upgrade process. An image that fails an integrity check will not be loaded.
FPT_STM.1	The TOE provides a source of date and time information used in audit event timestamps. The clock function is reliant on the system clock provided by the underlying hardware. The TOE can optionally be set to receive clock updates from an NTP server. This date and time is used as the time stamp that is applied to TOE generated audit records and used to track inactivity of administrative sessions.
FTA_SSL_EXT.1 FTA_SSL.3	A Security Administrator can configure maximum inactivity times for administrative sessions through the TOE GUI and CLI interfaces. The configuration of inactivity periods are applied on a per interface basis. A configured inactivity period will be applied to both local and remote sessions in the same manner. When the interface has been idle for more than the configured period of time, the session will be terminated and will require authentication to establish a new session.
FTA_SSL.4	A Security Administrator is able to exit out of both local and remote administrative sessions.
FTA_TAB.1	Security Administrators can define a custom login banner that will be displayed at the following interfaces, • Local CLI • Remote CLI • Remote GUI This banner will be displayed prior to allowing Security Administrator access through those interfaces.
FTP_ITC.1	The TOE supports communications with several types of authorized IT entities, including, • Audit Servers • LDAP Servers Each of these connections are protected via a TLS connection. This protects the data from disclosure by encryption using AES and by HMACs that verify that data has not been modified. TLS provides assured identification of the non-TSF endpoint by validating X.509 certificates. The TOE retains a trusted store of certificate authorities which it uses to verify digital signatures on those non-TSF certificates. The TOE is responsible for initiating the trusted channel with the external trusted IT entities.
FTP_TRP.1	All remote administrative communications take place over a secure encrypted session. Remote CLI connections take place over an SSHv2 tunnel. The SSHv2 session is encrypted using AES encryption to protect confidentiality and uses HMACs to protect integrity of traffic. Remote GUI connections take place over

	a TLS connection. The TLS session is encrypted using AES encryption and uses HMACs to protect integrity. The remote administrators can initiate both SSHv2 and TLS communications with the TOE.

Table 10 TOE Summary Specification SFR Description

6.1 Key Storage and Zeroization

The following table describes the origin, storage and zeroization of keys as relevant to FCS_CKM.4 and FPT_SKP_EXT.1 provided by the TOE.

Key	Type	Origin	Storage/Protection	Zeroization
Diffie Hellman private key	DH Key	TOE generated	RAM	Keys are overwritten with zeros when the session is closed.
Diffie Hellman public key	DH Key	TOE generated	RAM	Keys are overwritten with zeros when the session is closed.
SSH Private Key	RSA Private Key	TOE generated	ACL protected directory	Key is overwritten by zeros when the compliance declassify zeroize command is issued.
SSH Public Key	RSA Public Key	TOE generated	n/a - public	Key is overwritten by zeros when the compliance declassify zeroize command is issued.
SSH Session Key	AES Key	TOE generated	RAM	Keys are overwritten with zeros when the session is closed
TLS Private Key	RSA Private Key	TOE generated	ACL protected directory	Key is overwritten by zeros when the compliance declassify zeroize command is issued.
TLS Public Key	RSA Public Key	TOE generated	n/a - public	Key is overwritten by zeros when the compliance declassify zeroize command is issued.
TLS Session Encryption Key	AES Key	TOE generated	RAM	Keys are overwritten with zeros when the session is closed.
TLS Session Integrity Key	HMAC Key	TOE generated	RAM	Keys are overwritten with zeros when the session is closed.

Table 11 Key Storage & Zeroization

Annex A: References

The following documentation was used to prepare this ST:

Identifier	Description
[CC_PART1]	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
[CC_PART2]	Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated April 2017, version 3.1, Revision 5, CCMB-2017-004-002
[CC_PART3]	Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components April 2017, version 3.1, Revision 5, CCMB-2017-004-003
[CEM]	Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated April 2017, version 3.1, Revision 5, CCMB-2017-004-004
[NDcPP]	Collaborative Protection Profile for Network Devices, Version 2.0, 02 May 2017.
[SD]	Supporting Document Mandatory Technical Document: Evaluation Activities for Network Device cPP, Version 2.0, May-2017.
[800-38A]	NIST Special Publication 800-38A Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001
[800-38D]	NIST Special Publication 800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.
[800-56Ar2]	NIST Special Publication 800-56A Revision 2, May 2013, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography
[800-56B]	NIST Special Publication 800-56B Revision 1, September 2014, Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography
[FIPS PUB 186-4]	FIPS PUB 186-4 Federal Information Processing Standards Publication: Digital Signature Standard (DSS), July 2013.
[FIPS PUB 198-1]	FIPS PUB 198-1 Federal Information Processing Standards Publication: The Keyed-Hash Message Authentication Code (HMAC) July 2008
[800-90A]	NIST Special Publication 800-90A Revision 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 2015
[FIPS PUB 180-4]	FIPS PUB 180-4 Federal Information Processing Standards Publication Secure Hash Standard (SHS), August 2015.
[RFC3526]	RFC 3526, More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE), May 2003.
[RFC2818]	RFC 2818, HTTP Over TLS, May 2000.
[RFC4251]	RFC 4251, The Secure Shell (SSH) Protocol Architecture, January 2006.
[RFC4252]	RFC 4252, The Secure Shell (SSH) Authentication Protocol, January 2006.
[RFC4253]	RFC 4253, The Secure Shell (SSH) Transport Layer Protocol, January 2006.
[RFC4254]	RFC 4254, The Secure Shell (SSH) Connection Protocol January 2006.
[RFC5647]	RFC 5647, AES Galois Counter Mode for the Secure Shell Transport Layer Protocol, August 2009.
[RFC6668]	RFC 6668, SHA-2 Data Integrity Verification for the Secure Shell (SSH) Transport Layer Protocol, July 2012.
[RFC5246]	RFC 5246, The Transport Layer Security (TLS) Protocol Version 1.2, August 2008.
[RFC4346]	RFC 4346, The Transport Layer Security (TLS) Protocol Version 1.1, April 2006.
[RFC3268]	RFC 3268, Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security

	(TLS), June 2002.
[RFC5289]	RFC 5289, TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), August 2008.
[RFC6125]	RFC 6125, Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS), March 2011.
[RFC5280]	RFC 5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008.
[RFC6960]	RFC 6960, X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP, June 2013.
[RFC2986]	RFC 2986, PKCS #10: Certification Request Syntax Specification Version 1.7, November 2000.

Table 12: References