FortiWLM Wireless Manager 8.5

Security Target

Version 2.9

November 2021

Document prepared by

www.lightshipsec.com

Document History

Version	Date	Author	Description
1.0	25 Aug 2020	L Turner	Final for evaluation
1.1	26 Oct 2020	K Newton	Minor Updates
1.2	10 Nov 2020	L Turner	Address observations
1.3	26 Jan 2021	L Turner	Address observations
1.4	29 Jan 2021	K Newton	Update Technical Decisions
1.5	08 Feb 2021	K Newton	Minor Updates
1.6	17 Feb 2021	K Newton	Address observations
1.7	15 Mar 2021	G Nickel	Address observations
1.8	05 Apr 2021	K Newton	Address observations
1.9	06 July 2021	M Boire	Update Technical Decisions
2.0	17 Aug 2021	M Boire	Update Section 1.2, Identification
2.1	20 Aug 2021	G Nickel	Address lab comments
2.2	16 Sep 2021	K Newton	Address observations
2.3	27 Sep 2021	K Newton	Address observations
2.4	05 Oct 2021	K Newton	Address lab comments
2.5	08 Nov 2021	K Newton	Address evaluator comments
2.6	11 Nov 2021	G Nickel	Address Evaluator comments
2.7	15 Nov 2021	K Newton	Address lab comments
2.8	16 Nov 2021	K Newton	Address lab comments

Table of Contents

List of Tables

1 Introduction

1.1 Overview

1 This Security Target (ST) defines the Fortinet FortiWLM Wireless Manager 8.5 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

2 Fortinet's FortiWLM Wireless Manager 8.5 offers full management of Fortinet controllers and access points along with an extensive set of troubleshooting and reporting tools, all in a single pane of glass. The Wireless Manager offers the ability to see the status of your entire wireless network in one place, while also getting visibility into Spectrum, Wireless Intrusion, and other key wireless health statistics.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	Fortinet FortiWLM Wireless Manager 8.5 Build: 8.5-2fips-7
Security Target	Fortinet FortiWLM Wireless Manager 8.5 Security Target, v2.8

1.3 Conformance Claims

3 This ST supports the following conformance claims:

a) CC version 3.1 revision 5

b) CC Part 2 extended

c) CC Part 3 conformant

d) collaborative Protection Profile for Network Devices, v2.2e

e) NIAP Technical Decisions per Table 2

Table 2: NIAP Technical Decisions

TD #	Name	Rationale if n/a
TD0527	Updates to Certificate Revocation Testing (FIA_X509_EXT.1)	
TD0528	NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4	NTP not claimed
TD0536	NIT Technical Decision for Update Verification Inconsistency	
TD0537	NIT Technical Decision for Incorrect reference to FCS_TLSC_EXT.2.3	
TD0538	NIT Technical Decision for Outdated link to allowed-with list	
TD0546	NIT Technical Decision for DTLS - clarification of Application Note 63	DTLS not claimed

TD #	Name	Rationale if n/a
TD0547	NIT Technical Decision for Clarification on developer disclosure of AVA_VAN	
TD0555	NIT Technical Decision for RFC Reference incorrect in TLSS Test	
TD0556	NIT Technical Decision for RFC 5077 question	
TD0563	NIT Technical Decision for Clarification of audit date information	
TD0564	NIT Technical Decision for Vulnerability Analysis Search Criteria	
TD0569	NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7	DTLS not claimed
TD0570	NIT Technical Decision for Clarification about FIA_AFL.1	
TD0571	NIT Technical Decision for Guidance on how to handle FIA_AFL.1	
TD0572	NIT Technical Decision for Restricting FTP_ITC.1 to only IP address identifiers	
TD0580	NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e	
TD0581	NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3	
TD0591	NIT Technical Decision for Virtual TOEs and hypervisors	
TD0592	NIT Technical Decision for Local Storage of Audit Records	

1.4 Terminology

Table 3: Terminology

Term	Definition
CC	Common Criteria
EAL	Evaluation Assurance Level
NDcPP	collaborative Protection Profile for Network Devices
PP	Protection Profile
TOE	Target of Evaluation

Term	Definition
TSF	TOE Security Functionality

2 TOE Description

2.1 Type

4 The TOE is a network device that provides management of wireless controllers and access points.

2.2 Usage

2.2.1 Deployment

5 Figure 1 shows an example deployment of the TOE (enclosed in blue) in the context of a Fortinet controller-managed wireless network.

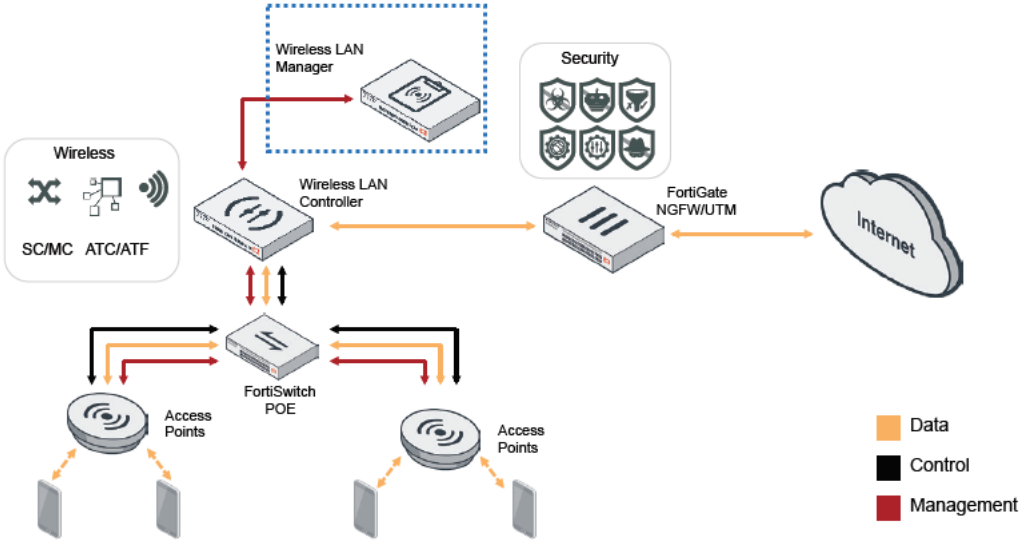

Figure 1: Example TOE deployment

2.2.2 Interfaces

6 The TOE interfaces are shown in Figure 2.

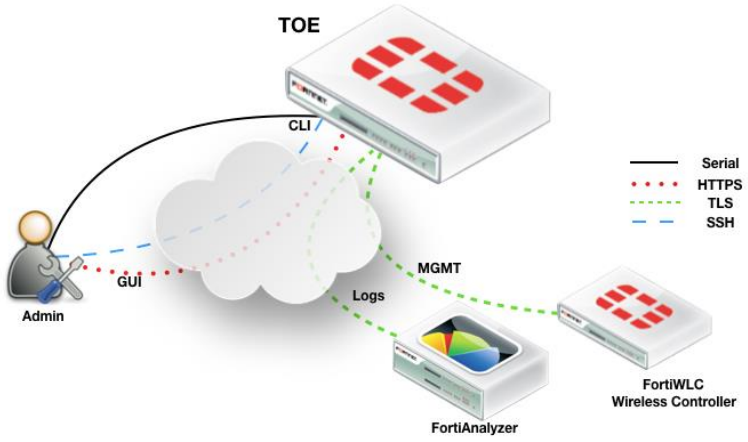

Figure 2: TOE interfaces

7 The TOE interfaces are as follows:

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

b) **GUI.** Administrative web GUI via HTTPS.

c) **Logs.** Forwarding of logs to a remote audit server, which is a Fortinet FortiAnalyzer, via TLS.

d) **MGMT.** Management of FortiWLC Wireless Controllers via TLS.

2.3 Security Functions

8 The TOE provides the following security functions:

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

b) **Cryptographic Support.** The TOE implements cryptographic libraries and protocols in support of its functions. Relevant Cryptographic Algorithm Validation Program (CAVP) certificates are shown in Table 4.

c) **Identification and Authentication.** The TOE implements authentication mechanisms, authentication failure handling, password management and X.509 certificate validation services.

d) **Security Management.** The TOE restricts the ability to manage its functions to Security Administrators.

e) **Protection of the TSF.** The TOE protects cryptographic keys and administrator passwords, performs a suite of self-tests and ensures the authenticity and integrity of software updates through digital signatures.

f) **TOE Access.** The TOE implements session locking, session termination and displays access banners.

g) **Trusted path/channels.** The TOE protects the integrity and confidentiality of communications as noted in section 2.2.2 above.

Table 4: CAVP Certificates

SFR	Capability	Key Size / Curve / Mod	Cryptographic Library	Certificate
FCS_CKM.1	RSA KeyGen (186-4)	2048	Fortinet FortiWLM SSL Cryptographic Library	C1653
	ECDSA KeyGen (186-4)	P-256		
	FFC KeyGen	(DH Group 14)		n/a
FCS_CKM.2	RSA (RFC 3447)	n/a		n/a
	KAS-ECC Component	P-256		C1653

SFR	Capability	Key Size / Curve / Mod	Cryptographic Library	Certificate
	FFC Schemes	(DH Group 14)	Fortinet FortiWLM SSL Cryptographic Library	n/a
FCS_COP.1 /DataEncryption	AES-CBC	128, 256		C1653
FCS_COP.1 /SigGen and SigVer	RSA SigGen (186-4) RSA SigVer (186-4)	2048		
FCS_COP.1 /Hash	SHA-1 SHA-256 SHA-384 SHA-512	160, 256, 384, 512		
FCS_COP.1 /KeyedHash	HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512	160, 256, 384, 512		
FCS_RBG_EXT.1	CTR_DRBG (AES)	-	Fortinet FortiWLM RBG Cryptographic Library	C1652

2.4 Physical Scope

9 The physical boundary of the TOE includes the Fortinet hardware equipped with Araneus Alea II USB Entropy Token and software which is delivered to the customer via commercial courier. *The software function that reads the entropy data periodically checks whether the inserted USB device has a vendor ID matching Araneus and a product ID matching the Alea token. If any other USB device is inserted, the TOE will refuse to recognize it and display an error via console.*

10 The TOE models in scope are shown in in Table 5.

Table 5: TOE models

Model	CPU	Target Deployment
FWM-100D	Intel Celeron J1900 (Bay Trail)	Small enterprise
FWM-1000D	Intel Core i7-4790S (Haswell)	Large enterprise

2.4.1 Guidance Documents

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

a) Fortinet FortiWLM Wireless Manager 8.5 FIPS140-2 and Common Criteria Technote

b) Fortinet FortiWLM Wireless Manager 8.5 User Guide

c) Fortinet FortiWLM Wireless Manager 8.5 Release Notes

d) Fortinet Hardware Guides:

i) Fortinet FortiWLM 100D QuickStart Guide

ii) Fortinet FortiWLM 1000D QuickStart Guide

2.4.2 Non-TOE Components

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

a) **Audit Server.** The TOE makes use of a FortiAnalyzer for remote logging.

b) **FortiWLC Wireless Controller.** The TOE manages Fortinet FortiWLC Wireless Controllers.

2.5 Logical Scope

13 The logical scope of the TOE comprises the security functions defined in section 2.3.

2.5.1 Functions not included in the TOE

14 For the TOE to be in the evaluated configuration, the following functions must not be enabled/used:

a) The Virtual Edition of the application suite

b) SNMP

c) Remote authentication (e.g. RADIUS, LDAP, TACACS+)

d) IPv6

e) Service Assurance Manager (SAM), Spectrum Manager, Wireless Intrusions Prevention System (WIPS)

f) Logging to syslog server

g) Logging to FortiCloud

3 Security Problem Definition

15 The Security Problem Definition is reproduced from section 4 of the NDcPP.

3.1 Threats

Table 6: Threats

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

Identifier	Description
	the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_ FUNCTIONALITY_ COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the Network Device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.
T.PASSWORD_ CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other Network Devices.
T.SECURITY_ FUNCTIONALITY_ FAILURE	An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

3.2 Assumptions

Table 7: Assumptions

Identifier	Description
A.PHYSICAL_ PROTECTION	The Network Device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP does not include any requirements on physical tamper protection or other physical attack mitigations. The cPP does not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device. For vNDs, this assumption applies to the physical platform on which the VM runs.
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). In the case of vNDs, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of the distributed TOE run inside more than one virtual machine (VM) on a single VS. There are no other guest VMs on the physical platform providing non-Network Device functionality.

Identifier	Description
A.NO_THRU_ TRAFFIC_ PROTECTION	A standard/generic Network Device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the Network Device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the Network Device, destined for another network entity, is not covered by the NDcPP. It is assumed that this protection will be covered by cPPs and PP-Modules for particular types of Network Devices (e.g., firewall).
A.TRUSTED_ ADMINISTRATOR	The Security Administrator(s) for the Network Device are assumed to be trusted and to act in the best interest of security for the organization. This includes appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The Network Device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are expected to fully validate (e.g. offline verification) any CA certificate (root CA certificate or intermediate CA certificate) loaded into the TOE's trust store (aka 'root store', ' trusted CA Key Store', or similar) as a trust anchor prior to use (e.g. offline verification).
A.REGULAR_ UPDATES	The Network Device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_ CREDENTIALS_ SECURE	The Administrator's credentials (private key) used to access the Network Device are protected by the platform on which they reside.
A.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.

3.3 Organizational Security Policies

Table 8: Organizational Security Policies

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

4 Security Objectives

16 The security objectives are reproduced from section 5 of the NDcPP.

Table 9: Security Objectives for the Operational Environment

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

5 Security Requirements

5.1 Conventions

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

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

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

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

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

e) **Iteration.** Indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash").

18 **Note:** Operations performed within the Security Target are denoted within brackets []. Operations shown without brackets are reproduced from the NDcPP.

5.2 Extended Components Definition

19 Refer to NDcPP.

5.3 Functional Requirements

Table 10: Summary of SFRs

Requirement	Title
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	Protected Audit Event Storage
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_HTTPS_EXT.1	HTTPS Protocol
FCS_RBG_EXT.1	Random Bit Generation

Requirement	Title
FCS_SSHS_EXT.1	SSH Server Protocol
FCS_TLSC_EXT.1	TLS Client protocol Without Mutual Authentication
FCS_TLSC_EXT.2	TLS Client Support for Mutual Authentication
FCS_TLSS_EXT.1/MGMT	TLS Server Protocol without Mutual Authentication
FCS_TLSS_EXT.2/MGMT	TLS Server Support for Mutual Authentication
FCS_TLSS_EXT.1/GUI	TLS Server Protocol without Mutual Authentication
FIA_AFL.1	Authentication Failure Management
FIA_PMG_EXT.1	Password Management
FIA_UIA_EXT.1	User Identification and Authentication
FIA_UAU_EXT.2	Password-based Authentication Mechanism
FIA_UAU.7	Protected Authentication Feedback
FIA_X509_EXT.1/Rev	X.509 Certificate Validation
FIA_X509_EXT.2	X.509 Certificate Authentication
FIA_X509_EXT.3	X.509 Certificate Requests
FMT_MOF.1/ManualUpdate	Management of Security Functions Behaviour
FMT_MOF.1/Functions	Management of Security Functions Behaviour
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
FPT_SKP_EXT.1	Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
FPT_APW_EXT.1	Protection of Administrator Passwords
FPT_TST_EXT.1	TSF Testing
FPT_TUD_EXT.1	Trusted Update
FPT_STM_EXT.1	Reliable Time Stamps

Requirement	Title
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination
FTA_TAB.1	Default TOE Access Banners
FTP_ITC.1	Inter-TSF trusted channel
FTP_TRP.1/Admin	Trusted Path

5.3.1 Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

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

a) Start-up and shutdown of the audit functions;

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

c) All administrative actions comprising:

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

d) *Specifically defined auditable events listed in **~~Table 2~~ Table 11**.*

Table 11: Audit Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
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_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_RBG_EXT.1	None.	None.
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_TLSC_EXT.2	None	None
FCS_TLSS_EXT.1/MGMT	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.2/MGMT	None	None
FCS_TLSS_EXT.1/GUI	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_UIA_EXT.1	All use of identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
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 • 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_MOF.1/Functions	None.	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_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_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session locking mechanism.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.

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

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

b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of* ***~~Table 2~~ Table 11***.

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 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 [overwrite previous audit records according to the following rule: [*overwrite oldest record first*], [*no other action*]] when the local storage space for audit data is full.

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

]~~and specified cryptographic key sizes [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: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 3447, “Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1”;
- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;
- FFC Schemes using “safe-prime” groups that meet the following: ‘NIST Special Publication 800-56A Revision 3, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography” and [RFC 3526]

] ~~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/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*] 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].

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

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

- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) [*2048 bits 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,]

FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

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

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

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

FCS_HTTPS_EXT.1 HTTPS Protocol

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

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

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

FCS_RBG_EXT.1 Random Bit Generation

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

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [[*one*] 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.

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, [5647, 5656, 6187, 6668].

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 [*262144*] 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].

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

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

FCS_SSHS_EXT.1.7 The TSF shall ensure that [diffie-hellman-group14-sha1] and [no other methods] 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 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_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_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 identifier per RFC 6125 section 6, 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 NIST curves/groups: [secp256r1] and no other curves] in the Client Hello.

FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

FCS_TLSC_EXT.2.1 The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

FCS_TLSS_EXT.1/MGMT TLS Server Protocol without Mutual Authentication

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

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268].

FCS_TLSS_EXT.1.2/MGMT 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/MGMT The TSF shall perform key establishment for TLS using [Diffie-Hellman parameters with size [2048 bits].

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

FCS_TLSS_EXT.2/MGMT TLS Server Support for Mutual Authentication

FCS_TLSS_EXT.2.1/MGMT The TSF shall support TLS communication with mutual authentication of TLS clients using X.509v3 certificates.

FCS_TLSS_EXT.2.2/MGMT When establishing a trusted channel, by default the TSF shall not establish a trusted channel if the client certificate is invalid. The TSF shall also [

- Not implement any administrator override mechanism].

FCS_TLSS_EXT.2.3/MGMT The TSF shall not establish a trusted channel if the identifier contained in a certificate does not match an expected identifier for the client. If the identifier is a Fully Qualified Domain Name (FQDN), then the TSF shall match the identifiers according to RFC 6125, otherwise the TSF shall parse the identifier from the certificate and match the identifier against the expected identifier of the client as described in the TSS.

FCS_TLSS_EXT.1/GUI TLS Server Protocol

FCS_TLSS_EXT.1.1/GUI 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:[

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_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/GUI The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [none].

FCS_TLSS_EXT.1.3/GUI The TSF shall perform key establishment for TLS using [Diffie-Hellman parameters with size [2048 bits], ECDHE curves [secp256r1] and no other curves].

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

Application Note: TLS 1.1 supports a subset of the above suites (it does not support SHA256 or SHA384).

5.3.3 Identification and Authentication (FIA)

FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*1-5*] 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 remote session using any authentication method that involves a password until an Administrator defined time period has elapsed].

FIA_PMG_EXT.1 Password Management

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “-”, “,”, “<”, “=”, “>”, “?”, “\”, “”, “+”, “-“, “.”, “/”, “_”];

b) Minimum password length shall be configurable to between [*8*] and [*15] characters.*

FIA_UIA_EXT.1 User Identification and Authentication

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

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

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

FIA_UAU_EXT.2 Password-based Authentication Mechanism

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

FIA_UAU.7 Protected Authentication Feedback

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

FIA_X509_EXT.1/Rev X.509 Certificate Validation

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

- RFC 5280 certificate validation and certification path validation **supporting a minimum path length of three certificates**.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.

- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the presence of the basicConstraints extension and that the CA flag is set to TRUE.
- The TSF shall validate the revocation status of the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 6960]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
    - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
    - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
    - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

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

FIA_X509_EXT.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 as specified by RFC 2986 and be able to provide the following information in the request: public key and [Common Name, Organization, Organizational Unit, Country].

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

5.3.4 Security Management (FMT)

FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

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

FMT_MOF.1/Functions Management of Security Functions Behaviour

FMT_MOF.1.1/Functions The TSF shall restrict the ability to [modify the behaviour of] the functions [transmission of audit data to an external IT entity] to *Security Administrators*.

FMT_MTD.1/CoreData Management of TSF Data

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

FMT_MTD.1/CryptoKeys Management of TSF data

FMT_MTD.1.1/CryptoKeys The TSF shall restrict the ability to *manage* the *cryptographic keys to Security Administrators*.

FMT_SMF.1 Specification of Management Functions

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- [
    - Ability to configure audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full);
    - Ability to manage the cryptographic keys;
    - Ability to configure thresholds for SSH rekeying;
    - Ability to set the time which is used for time-stamps;
    - 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;
    - No other capabilities].

FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- *Security Administrator*.

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

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

are satisfied.

5.3.5 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 administrative passwords in non-plaintext form.

FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext administrative 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: [

- *Firmware integrity tests*
- *Configuration integrity tests*
- *Cryptographic algorithm tests*
- *DRGB tests*
- *BIOS tests*
- *Boot loader image verification*].

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 [no other TOE firmware/software version].

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

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

FPT_STM_EXT.1 Reliable Time Stamps

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

FPT_STM_EXT.1.2 The TSF shall [allow the Security Administrator to set the time].

5.3.6 TOE Access (FTA)

FTA_SSL_EXT.1 TSF-initiated Session Locking

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

- terminate the session]

after a Security Administrator-specified time period of inactivity.

FTA_SSL.3 TSF-initiated Termination

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

FTA_SSL.4 User-initiated Termination

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

FTA_TAB.1 Default TOE Access Banners

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

5.3.7 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, [FortiWLC]** 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, FortiWLC management*].

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*.

5.4 Assurance Requirements

20 The TOE security assurance requirements are summarized in Table 12.

Table 12: Assurance Requirements

Assurance Class	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 Procedures
Life Cycle Support	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests	ATE_IND.1	Independent Testing - conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Survey

21 In accordance with section 7.1 of the NDcPP, the following refinement is made to ASE:

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

6 TOE Summary Specification

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

6.1 Security Audit

6.1.1 FAU_GEN.1

23 The TOE generates the audit records specified at FAU_GEN.1 containing the following fields:

a) Log number

b) **Date/Time.** The time that the log message was created.

c) **Level.** The level of the log message. The available logging levels are:

i) Alert: Immediate action is required.

ii) Critical: Functionality is affected.

iii) Error: Functionality is probably affected.

iv) Warning: Functionality might be affected.

v) Information: Information about normal events.

vi) Debug: Information used for diagnosis or debugging.

d) **User.** The user to which the log message relates. User can be a specific user or system.

e) **Message.** Detailed log message.

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

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

b) **Generate CSR.** Action and key reference.

c) **Import Certificate.** Action and key reference.

d) **Import CA Certificate.** Action and key reference.

6.1.2 FAU_GEN.2

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

6.1.3 FAU_STG_EXT.1

26 The Security Administrator configures the TOE to send logs to a FortiAnalyzer. Log events are sent in real-time. Logs are sent via TLS.

27 The amount of audit data that may be stored locally is dependent on the available disk space which varies depending on TOE model.

28 When the local audit data store is full, the TOE will overwrite audit records starting with the oldest audit record.

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

6.2 Cryptographic Support

6.2.1 FCS_CKM.1

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

a) **RSA 2048-bit.** Used in SSH and TLS RSA ciphersuites.

b) **ECC P-256.** Used in TLS ECC ciphersuites.

c) **Diffie-Hellman Group 14.** Diffie-Hellman used in TLS and SSH.

6.2.2 FCS_CKM.2

31 The TOE supports the following key establishment schemes:

a) **RSA schemes.** Used in TLS ciphersuites with RSA key exchange. TOE is both sender and receiver.

b) **ECC schemes.** Used in TLS ciphersuites with ECDH key exchange. TOE is both sender and receiver.

c) **Diffie-Hellman Group 14.** Used in TLS and SSH. The TOE meets RFC 3526 Section 3 by implementing the 2048-bit Modular Exponential (MODP) Group.

6.2.3 FCS_CKM.4

32 All persistent private keys are encrypted. Keys held in volatile memory are zeroized after use by overwriting the key storage area with zeroes. Keys held in flash memory may be destroyed using a Command Line Interface (CLI) command to overwrite the entire flash memory with zeroes. This command is used when a device is reset or taken out of operation. Table 14 shows the origin, storage location and destruction details for cryptographic keys and passwords. Unless otherwise stated, the keys are generated by the TOE.

6.2.4 FCS_COP.1/DataEncryption

33 The TOE provides symmetric encryption and decryption capabilities using 128 and 256 bit AES in CBC mode. AES is implemented in the following protocols: TLS and SSH (CBC only).

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

6.2.5 FCS_COP.1/SigGen

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

a) RSA Signature Algorithm with key size of 2048 and greater,

36 RSA signature verification services are used in the TLS protocols. Additionally, RSA signature verification is used for the SSH protocol (ssh-rsa) and TOE firmware integrity checks.

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

6.2.6 FCS_COP.1/Hash

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

39 SHS is implemented in the following parts of the TSF:

a) TLS and SSH;

b) Digital signature verification as part of trusted update validation; and

c) Hashing of passwords in non-volatile storage.

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

6.2.7 FCS_COP.1/KeyedHash

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

42 HMAC is implemented in the following protocols: TLS and SSH.

43 The characteristics of the HMACs used in the TOE are given in Table 13.

Table 13: HMAC Characteristics

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

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

6.2.8 FCS_HTTPS_EXT.1

45 The TOE web GUI is accessed via an HTTPS connection using the TLS implementation described by FCS_TLSS_EXT.1. The TOE does not use HTTPS in a client capacity. The TOE's HTTPS protocol complies with RFC 2818.

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

6.2.9 FCS_RBG_EXT.1

47 The TOE contains a CTR_DRBG that is seeded from the hardware entropy source. Entropy from the noise source is extracted 5120 bits at a time, conditioned and used to seed the DRBG with 256 bits of full entropy.

48 Additional detail is provided the proprietary Entropy Description.

6.2.10 FCS_SSHS_EXT.1

49 The TOE implements SSH in compliance with RFCs 4251 through 4254, 5647, 5656, 6187 and 6668.

50 The TOE supports password-based or public key (ssh-rsa, rsa-sha2-256, rsa-sha2-512) authentication.

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

52 The TOE utilises AES-CBC-128 and AES-CBC-256 for SSH encryption.

53 The TOE provides data integrity for SSH connections via HMAC-SHA1, HMAC-SHA2-256 and HMAC-SHA2-512.

54 The TOE supports Diffie-Hellman Group 14 SHA-1 (diffie-hellman-group14-sha1) for SSH key exchanges.

55 The TOE will re-key SSH connections after 1 hour of after an aggregate of 1 gig of data has been exchanged (whichever occurs first).

56 The TOE supports SSH authentication using RSA Keys. The administrator must first import the RSA public key for the admin account using the CLI. Please follow configuration steps shown in the guidance document for the same.

6.2.11 FCS_TLSC_EXT.1

57 The TOE operates as a TLS client for the trusted channel with the FortiAnalyzer audit server.

58 Only TLS 1.2 protocol version is allowed and ciphersuites are restricted to the following:

a) TLS_DHE_RSA_WITH_AES_128_CBC_SHA

b) TLS_DHE_RSA_WITH_AES_256_CBC_SHA

c) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

d) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

e) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

f) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

59 Ciphersuites are not user-configurable.

60 The reference identifiers for the FortiAnalyzer Server are configured by the administrator using the web GUI. The reference identifiers must be a DNS name.

61 When the TLS client receives an X.509 certificate from the server, 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 (DNS name) in the certificate, then the TOE 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. For DNS Name matching, the hostname must be an exact match or wildcard match. In the case of a wildcard match; the wildcard must be the left-most component, wildcard matches a single component, and there are at least two non-wildcard components

62 The TLS client does not support certificate pinning.

63 The TLS client will transmit the Supported Elliptic Curves extension in the Client Hello message by default with support for the following NIST curves: P256. The

non-TOE server can choose to negotiate the elliptic curve from this set for any of the mutually negotiable elliptic curve ciphersuites.

6.2.12 FCS_TLSC_EXT.2

64 The TOE supports mutual authentication using X.509v3 certificates when establishing TLS sessions.

65 The TLS client will transmit its leaf certificate to the server as required by FortiWLC in support of the mutual authentication process.

6.2.13 FCS_TLSS_EXT.1&2/MGMT

66 The TOE operates as a TLS server for the trusted channel with FortiWLC controllers.

67 The server only allows TLS protocol version 1.2 (rejecting any other protocol version) and is restricted to the following ciphersuites by default:

a) TLS_DHE_RSA_WITH_AES_256_CBC_SHA

68 Ciphersuites are not user-configurable.

69 Session tickets are supported and follow the structure defined by RFC 5077. Tickets are protected using AES128 CBC and HMAC SHA256. Keys are randomly generated for both algorithms.

70 The TLS server is capable of negotiating ciphersuites that include DHE, and ECDHE key agreement schemes. The DHE key agreement parameters are restricted to 2048 bits and are hardcoded into the server.

71 When the TLS server receives an X.509 certificate from the client, the server 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 (DNS name) in the certificate, then the TOE 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.

72 The TOE does not support any fallback authentication functions.

6.2.14 FCS_TLSS_EXT.1/GUI

73 The TOE operates as a TLS server for the web GUI trusted path.

74 The server only allows TLS protocol versions 1.1 and 1.2 (rejecting any other protocol version).

75 When using TLS 1.1, following ciphersuites are supported:

a) TLS_DHE_RSA_WITH_AES_128_CBC_SHA

b) TLS_DHE_RSA_WITH_AES_256_CBC_SHA

c) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

d) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

76 When using TLS 1.2, following ciphersuites are supported:

a) TLS_DHE_RSA_WITH_AES_128_CBC_SHA

b) TLS_DHE_RSA_WITH_AES_256_CBC_SHA

c) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

d) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

e) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256

f) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384

77 Ciphersuites are not user-configurable.

78 Session tickets are supported and follow the structure defined in RFC 5077. Tickets are protected using AES128 CBC and HMAC SHA256. Keys are randomly generated for both algorithms.

79 The TLS server is capable of negotiating ciphersuites that include DHE, and ECDHE key agreement schemes. The DHE key agreement parameters are restricted to 2048 bits and are hardcoded into the server.

6.3 Identification and Authentication

6.3.1 FIA_PMG_EXT.1

80 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 “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”.

81 The minimum password length is settable by the Administrator and can range from 8 to 15 characters.

6.3.2 FIA_UIA_EXT.1

82 The TOE requires all users to be successfully identified and authenticated. The TOE warning banner may be viewed prior to authentication.

83 Administrative access to the TOE is facilitated through one of several interfaces:

a) Directly connecting to the TOE appliance

b) Remotely connecting to each appliance via SSHv2

c) Remotely connecting to appliance GUI via HTTPS

6.3.3 FIA_UAU_EXT.2

84 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.

85 The TOE provides a local password based authentication mechanism.

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

6.3.4 FIA_UAU.7

87 For all authentication at the local CLI the TOE displays no characters when the administrative password is entered so that the password is obscured.

6.3.5 FIA_AFL.1

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

89 When a user account has sequentially failed authentication the configured number of times (default 5), the account will be locked for a Security Administrator defined time period (default 5 minutes).

90 The administrator can configure the maximum number of failed attempts using the web GUI or CLI.

91 The local console does not implement the lockout mechanism.

6.3.6 FIA_X509_EXT.1/Rev

92 The TOE performs X.509 certificate validation at the following points:

a) TOE TLS client validation of server X.509 certificates;

b) 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).

93 In all scenarios, certificates are checked for several validation characteristics:

a) If the certificate ‘notAfter’ date is in the past, then this is an expired certificate which is considered invalid;

b) The certificate chain must terminate with a trusted CA certificate;

c) Server certificates consumed by the TOE TLS client must have a ‘serverAuthentication’ extendedKeyUsage purpose;

d) 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.

94 Certificate revocation checking for the above scenarios is performed using OCSP.

95 As X.509 certificates are not used for trusted updates, firmware integrity self-tests or client authentication, the code-signing and clientAuthentication purpose is not checked in the extendedKeyUsage for related certificates.

96 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:

a) The public key algorithm and parameters are checked

b) The current date/time is checked against the validity period revocation status is checked

c) Issuer name of X matches the subject name of X+1

d) Name constraints are checked

e) Policy OIDs are checked

f) Policy constraints are checked; issuers are ensured to have CA signing bits

g) Path length is checked

h) Critical extensions are processed

97 If, during the entire trust chain verification activity, any certificate under review fails a verification check, then the entire trust chain is deemed untrusted.

6.3.7 FIA_X509_EXT.2

98 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.

99 Instructions for configuring the trusted IT entities to supply appropriate X.509 certificates are captured in the guidance documents.

100 As part of the verification process, OCSP is used to determine whether the certificate is revoked or not. If the OCSP responder cannot be contacted, then the TOE will accept the certificate.

6.3.8 FIA_X509_EXT.3

101 For the Certificate Signing Request, a CN is required and may be an IP address, DNS name or email address. SANs are optional and may be email, IP address, URI, DNS name or directory name.

6.4 Security Management

6.4.1 FMT_MOF.1/ManualUpdate

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

6.4.2 FMT_MOF.1/Functions

103 The TOE restricts the ability to modify (enable/disable) transmission of audit records to an external audit server to Security Administrators.

6.4.3 FMT_MTD.1/CoreData

104 Users are required to login before being provided with access to any administrative functions.

6.4.4 FMT_SMR.2

105 The TOE operates two pre-defined users:

a) **Admin.** Privileged access to the TOE. This profile equates to the Security Administrator role defined in this Security Target.

106 Management of TSF data via the CLI or web GUI is restricted to Security Administrators.

6.4.5 FMT_MTD.1/CryptoKeys

107 The TOE restricts the ability to manage SSH, TLS and any configured X.509 private keys to Security Administrators.

6.4.6 FMT_SMF.1

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

a) Ability to administer the TOE locally and remotely

b) Ability to configure the access banner

c) Ability to configure the session inactivity time before session termination or locking

d) Ability to update the TOE and to verify the updates

e) Ability to configure the authentication failure parameters

f) Ability to configure audit behavior (enable/disable remote logging)

g) Ability to manage the cryptographic keys

h) Ability to configure thresholds for SSH rekeying

i) Ability to set the time which is used for time-stamps

j) Ability to import X.509v3 certificates to the TOE's trust store

6.5 Protection of the TSF

6.5.1 FPT_SKP_EXT.1

109 Keys are protected as described in Table 14. In all cases, plaintext keys cannot be viewed through an interface designed specifically for that purpose.

Table 14: Keys

Key/Password	Generation/ Algorithm	Storage	Zeroization
TLS Private Key	RSA (2048 bits)	Flash - plaintext	Overwritten with zeroes by erase-disk command.
TLS Public Key	RSA (2048 bits)	Flash - plaintext	n/a – public key
DH Keys used for TLS	DH (2048 bits / P-256)	RAM - plaintext	Overwritten with zeroes upon termination of the session or reboot of the appliance
AES key used for TLS	AES-128 AES-256	RAM - plaintext	Overwritten with zeroes upon termination of the session or reboot of the appliance
Firmware Update Key (Public Key)	Preconfigured RSA (2048 bits)	Flash - plaintext	n/a – public key
SSH Private Key (host key)	RSA (2048 bits)	Flash - plaintext	Overwritten with zeroes by erase-disk command.
SSH Public Key	RSA (2048 bits)	Flash - plaintext	n/a – public key

Key/Password	Generation/ Algorithm	Storage	Zeroization
SSH Session Key	AES-128 AES-256	RAM - plaintext	The keys (including re-keyed keys) are overwritten with zeroes when no longer required or reboot of the appliance

6.5.2 FPT_APW_EXT.1

110 Passwords are protected as describe in Table 15. In all cases plaintext passwords cannot be viewed through an interface designed specifically for that purpose.

Table 15: Passwords

Key/Password	Generation/ Algorithm	Storage	Zeroization
Locally stored administrator passwords	User generated	Flash - SHA-512 hash	Overwritten with zeroes by erase-disk command.

6.5.3 FPT_TST_EXT.1

111 At startup, the TOE undergoes the following tests:

a) Firmware integrity test using SHA-256

b) Configuration integrity test using HMAC SHA-256

c) AES, CBC mode, encrypt known answer test

d) AES, CBC mode, decrypt known answer test

e) HMAC SHA-1 known answer test

f) SHA-1 known answer test (tested as part of HMAC SHA-1 known answer test)

g) HMAC SHA-256 known answer test

h) SHA-256 known answer test (tested as part of HMAC SHA-256 known answer test)

i) HMAC SHA-512 known answer test

j) SHA-512 known answer test (tested as part of HMAC SHA-512 known answer test)

k) RSA signature generation known answer test

l) RSA signature verification known answer test

m) ECDSA signature generation known answer test

n) ECDSA signature verification known answer test

o) DRBG known answer test

p) Central Processing Unit (CPU) and Memory Basic Input/Output System (BIOS) self-tests – CPU and memory are initialized by exercising a set of known answer tests and the BIOS is compared against a known checksum of

the image. The memory is zeroized and then a random pattern is written to and read from the memory.

q) Boot loader image verification – the boot loader compares the image of the TOE to a known checksum of the image prior to booting.

112 These tests ensure the correct operation of the cryptographic functionality of the TOE, the CPU and BIOS and verify that the correct TOE image is being used. The cryptographic functionality will not be available if the tests fail, and any operation of the TOE supported by this functionality will not be available. If the CPU, or BIOS tests fail, the device will not complete the boot up operation. If the boot loader image verification fails, the boot up operation will fail. When the device completes the boot up operation, this is evidence that the self-tests have passed, and that the TOE, and the cryptographic functions are operating correctly.

113 If a self-test fails, the device enters error mode and halts system operation. All data output and cryptographic services are inhibited when in the error state. Continued operation indicates that the tests have passed, and the TOE is operating correctly.

6.5.4 FPT_TUD_EXT.1

114 The current firmware version may be queried using the CLI or the web UI.

115 The administrator downloads firmware updates from Fortinet and manually installs the update.

116 The process to upload the firmware image file includes verification of the digital signature using the Fortinet firmware update key (RSA 2048), which is held in an encoded format in the current firmware image. If the digital signature on the firmware image cannot be verified, an error message appears. Once uploaded and verified, the new image is automatically installed; then the TOE will reboot and the firmware image is automatically activated.

6.5.5 FPT_STM_EXT.1

117 The TOE incorporates an internal clock that is used to maintain date and time. The Security Administrator sets the date and time during initial TOE configuration and may change the time during operation.

118 The TOE makes used of time for the following:

a) Audit record timestamps

b) Session timeouts (lockout enforcement)

c) Certificate validation

6.6 TOE Access

6.6.1 FTA_SSL_EXT.1

119 The Security Administrator may configure the TOE to terminate an inactive local interactive session (CLI) following a specified period of time.

6.6.2 FTA_SSL.3

120 The Security Administrator may configure the TOE to terminate an inactive remote interactive session (CLI and Web UI) following a specified period of time.

6.6.3 FTA_SSL.4

121 Administrative users may terminate their own sessions at any time using the exit command.

6.6.4 FTA_TAB.1

122 The TOE displays an administrator configurable message to users prior to login at the CLI and web GUI.

6.7 Trusted Path/Channels

6.7.1 FTP_ITC.1

123 The TOE supports secure communication with the following IT entities:

a) Audit server per FCS_TLSC_EXT.1

b) FortiWLC Wireless Controllers per FCS_TLSS_EXT.1/MGMT

6.7.2 FTP_TRP.1/Admin

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

a) CLI over SSH per FCS_SSHS_EXT.1

b) Web GUI over HTTPS per FCS_HTTPS_EXT.1.1

7 Rationale

7.1 Conformance Claim Rationale

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

a) **TOE type.** As identified in section 2.1, the TOE is network device, consistent with the NDcPP.

b) **Security problem definition.** As shown in section 3, the threats, OSPs and assumptions are reproduced directly from the NDcPP.

c) **Security objectives.** As shown in section 4, the security objectives are reproduced directly from the NDcPP.

d) **Security requirements.** As shown in section 5, the security requirements are reproduced directly from the NDcPP. No additional requirements have been specified.

7.2 Security Objectives Rationale

126 All security objectives are drawn directly from the NDcPP.

7.3 Security Requirements Rationale

127 All security requirements are drawn directly from the NDcPP. Table 16 presents a mapping between threats and SFRs as presented in the NDcPP.

Table 16: NDcPP SFR Rationale

Identifier	SFR Rationale
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	• The Administrator role is defined in FMT_SMR.2 and the relevant administration capabilities are defined in FMT_SMF.1 and FMT_MTD.1/CoreData, with optional additional capabilities in FMT_MOF.1/Services and FMT_MOF.1/Functions • The actions allowed before authentication of an Administrator are constrained by FIA_UIA_EXT.1, and include the advisory notice and consent warning message displayed according to FTA_TAB.1 • The requirement for the Administrator authentication process is described in FIA_UAU_EXT.2 • Locking of Administrator sessions is ensured by FTA_SSL_EXT.1 (for local sessions), FTA_SSL.3 (for remote sessions), and FTA_SSL.4 (for all interactive sessions) • The secure channel used for remote Administrator connections is specified in FTP_TRP.1/Admin • (Malicious actions carried out from an Administrator session are separately addressed by T.UNDETECTED_ACTIVITY)

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

Identifier	SFR Rationale
T.UPDATE_COMPROMISE	• Requirements for protection of updates are set in FPT_TUD_EXT.1 • Additional optional use of certificate-based protection of signatures can be specified using FPT_TUD_EXT.2, supported by the X.509 certificate processing requirements in FIA_X509_EXT.1, FIA_X509_EXT.2 and FIA_X509_EXT.3 • Requirements for management of updates are defined in FMT_SMF.1 and (for manual updates) in FMT_MOF.1/ManualUpdate, with optional requirements for automatic updates in FMT_MOF.1/AutoUpdate
T.UNDETECTED_ACTIVITY	• Requirements for basic auditing capabilities are specified in FAU_GEN.1 and FAU_GEN.2, with timestamps provided according to FPT_STM_EXT.1 • Requirements for protecting audit records stored on the TOE are specified in FAU_STG.1 • Requirements for secure transmission of local audit records to an external IT entity via a secure channel are specified in FAU_STG_EXT.1 • Optional additional requirements for dealing with potential loss of locally stored audit records are specified in FAU_STG_EXT.2/LocSpace, and FAU_STG.3/LocSpace • If (optionally) configuration of the audit functionality is provided by the TOE then this is specified in FMT_SMF.1, and confining this functionality to Security Administrators is required by FMT_MOF.1/Functions.
T.SECURITY_FUNCTIONALITY_COMPROMISE	• Protection of secret/private keys against compromise is specified in FPT_SKP_EXT.1 • Secure destruction of keys is specified in FCS_CKM.4 • If (optionally) management of keys is provided by the TOE then this is specified in FMT_SMF.1, and confining this functionality to Security Administrators is required by FMT_MTD.1/CryptoKeys • (Protection of passwords is separately covered under T.PASSWORD_CRACKING)
T.PASSWORD_CRACKING	• Requirements for password lengths and available characters are set in FIA_PMG_EXT.1 • Protection of password entry by providing only obscured feedback is specified in FIA_UAU.7 • Actions on reaching a threshold number of consecutive password failures are specified in FIA_AFL.1

Identifier	SFR Rationale
	• Requirements for secure storage of passwords are set in FPT_APW_EXT.1.
T.SECURITY_FUNCTIONALITY_FAILURE	• Requirements for running self-test(s) are defined in FPT_TST_EXT.1 • Optional use of certificates to support self-test(s) is defined in FPT_TST_EXT.2 (with support for the use of certificates in FIA_X509_EXT.1, FIA_X509_EXT.2, and FIA_X509_EXT.3)