FortiManager 6.2.8

Security Target

Version 1.7

November 2022

Document prepared by

www.lightshipsec.com

Document History

Version	Date	Author	Description
1.0	07 Jan 2022	M Ibrishimova	Addressed CB OR
1.1	12 Jan 2022	M Ibrishimova	Addressed OR07
1.2	22 Mar 2022	M Ibrishimova	Addressed OR06 and OR08
1.3	04 May 2022	M Ibrishimova	Addressed OR09
1.4	21 July 2022	M Ibrishimova	Updated build.
1.5	16 Aug 2022	M Ibrishimova	Added new TD.
1.6	18 Oct 2022	M Ibrishimova	Addressed OR10
1.7	6 Nov 2022	M Ibrishimova	Addressed OR11

Table of Contents

List of Tables

1 Introduction

1.1 Overview

89 This Security Target (ST) defines the FortiManager 6.2.8 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.

90 Fortinet FortiManager allows enterprises to manage all off their Fortinet devices in a single console central management system. FortiManager provides full visibility of your network, offering streamlined provisioning and innovative automation tools.

1.2 Identification

Table 1: Evaluation identifiers

Target of Evaluation	FortiManager 6.2.8 Build: v6.2.8-build9589
Security Target	FortiManager 6.2.8 Security Target, v1.7

1.3 Conformance Claims

91 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	The TOE does not claim FCS_NTP_EXT.1
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TD0538	NIT Technical Decision for Outdated link to allowed-with list	

TD #	Name	Rationale if n/a
TD0546	NIT Technical Decision for DTLS - clarification of Application Note 63	The TOE does not claim FCS_DTLS_EXT.1
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	
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	
TD0631	NIT Technical Decision for Clarification of public key authentication for SSH Server	
TD0632	NIT Technical Decision for Consistency with Time Data for vNDs	
TD0633	NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance	The TOE does not claim FCS_IPSEC_EXT.1

TD #	Name	Rationale if n/a
TD0634	NIT Technical Decision for Clarification required for testing IPv6	
TD0635	NIT Technical Decision for TLS Server and Key Agreement Parameters	
TD0636	NIT Technical Decision for Clarification of Public Key User Authentication for SSH	The TOE does not claim FCS_SSHC_EXT.1
TD0638	NIT Technical Decision for Key Pair Generation for Authentication	
TD0639	NIT Technical Decision for Clarification for NTP MAC Keys	The TOE does not claim NTP
TD0670	NIT Technical Decision for Mutual and Non-Mutual Auth TLSC Testing	

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
TSF	TOE Security Functionality

2 TOE Description

2.1 Type

92 The TOE is a network device that provides centralized management of other Fortinet devices.

2.2 Usage

2.2.1 Deployment

93 Figure 1 depicts an example deployment of the TOE (enclosed in red). The TOE is deployed to manage Fortinet products, including firewalls, FortiAnalyzers, switches, wireless infrastructure and Endpoints.

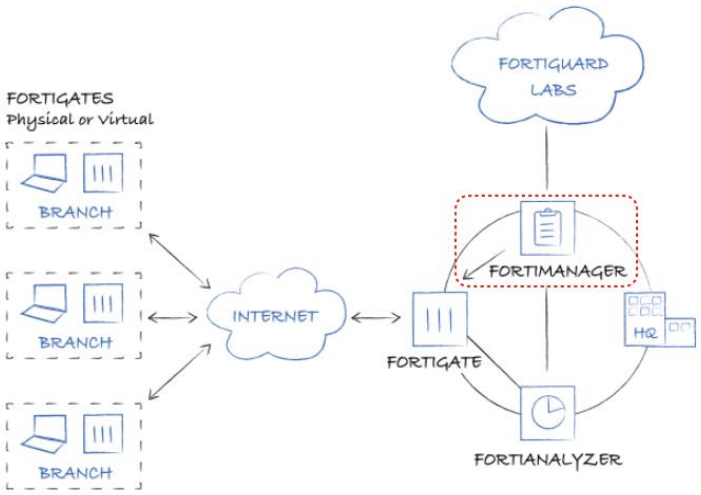


Figure 1: Example TOE deployment

2.2.2 Interfaces

94 The TOE management interfaces are shown in Figure 2.

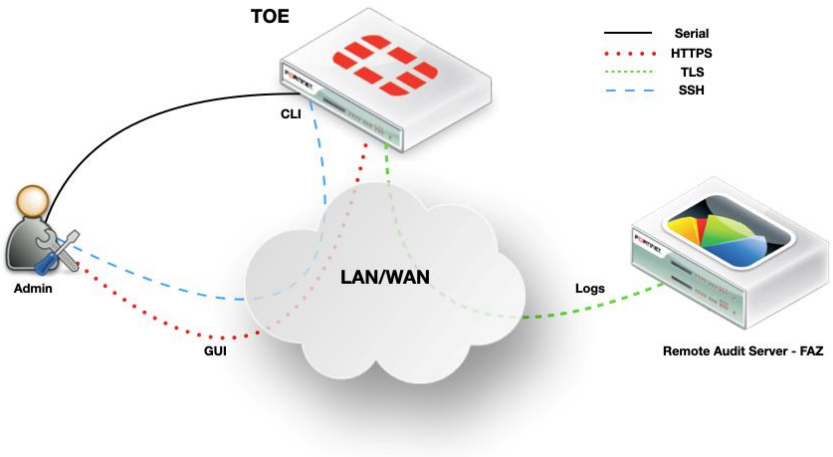


Figure 2: TOE interfaces

95 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 TOE audit events to a remote audit server, which is a Fortinet FortiAnalyzer, via TLS.

2.3 Security Functions / Logical Scope

96 The TOE provides the following security functions:

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

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

i) Administrator authentication with passwords

ii) Configurable password policies

iii) Role Based Access Control

iv) Access banners

v) Management of critical security functions and data

vi) Protection of cryptographic keys and passwords

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

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

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

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

Table 4: CAVP Certificates

Algorithm Capability	Certificate
AES GCM	C1987 A1063 C1984 C1907 A1963
AES CBC	
AES CTR	
SHA1, SHA2-256, SHA2-384, SHA2-512	
RSA KeyGen (186-4)	
RSA SigGen (186-4)	
RSA SigVer (186-4)	

Algorithm Capability	Certificate
ECDSA KeyGen (186-4)	
ECDSA SigGen (186-4)	
ECDSA SigVer (186-4)	
HMAC-SHA1, HMAC-SHA2-256, HMAC-SHA2-384, HMAC-SHA2-512	
CTR_DRBG	
KAS-ECC Component, KAS-ECC-SSC Sp800-56Ar3	
KAS-FFC Component, KAS-FFC-SSC Sp800-56Ar3	

2.4 Physical Scope

97 The physical boundary of the TOE includes all software and hardware shown in Table 5. The TOE is delivered via commercial courier.

Table 5: TOE hardware models

Model	CPU	Entropy	Storage
FMG-300F	Intel i3-6100 (Skylake)	Araneus Alea II Entropy Token	4 x 4TB
FMG-1000F	Intel Xeon Bronze 3106 (Skylake)	Araneus Alea II Entropy Token	8 x 4TB

2.4.1 Guidance Documents

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

a) FortiManager 6.2 NDcPP Common Criteria and FIPS 140-2 Technote, 02-628-740959-20210818

b) FortiManager 300F QuickStart Guide, 02-560-443236-20200828

c) FortiManager 1000F QuickStart Guide, 02-604-559470-20210512

d) FortiManager – CLI Reference v 6.2.8, 02-628-539012-20210513

e) FortiManager – Administration Guide v 6.2.8, 02-628-476230-20211022

99 Guides are available at: https://docs.fortinet.com/fortimanager/

2.4.2 Non-TOE Components

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

a) **Audit Server.** The TOE sends audit events to a Fortinet FortiAnalyzer.

b) **CRL Server.** The TOE uses a CRL server for certificate management.

2.4.3 Functions not included in the TOE Evaluation

101 The evaluation scope is limited to the security functions identified in section 2.3. While FortiManager can act as a FortiAnalyzer, this behavior is not in scope for this evaluation. In particular, FortiManager's log aggregation functionality (plaintext or TLS-protected) is not included in the scope of the evaluation. Product functionality such as security automation, and provisioning have also not been evaluated. The REST API has not been evaluated and has been disabled in the evaluated configuration.

102 The table below lists the TOE hardware models that were not evaluated.

Table 6: Vendor affirmed hardware

Model	CPU	Entropy	Storage
FMG-3000F	Intel Xeon E5-2630v3 (Haswell)	Araneus Alea II Entropy Token	16 x 3TB
FMG-3700F	Intel Xeon E5-2640V4 (Broadwell)	Araneus Alea II Entropy Token	60 x 4TB + 6 x 1.6TB NVMe SSD

3 Security Problem Definition

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

3.1 Threats

Table 7: 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 8: Assumptions

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

Identifier	Description
A.LIMITED_ FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality). If a virtual TOE evaluated as a pND, following Case 2 vNDs as specified in Section 1.2, the VS is considered part of the TOE with only one vND instance for each physical hardware platform. The exception being where components of a distributed TOE run inside more than one virtual machine (VM) on a single VS. In Case 2 vND, no non-TOE guest VMs are allowed on the platform. Application Note: Changed by TD0591
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.

Identifier	Description
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 9: 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

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

Table 10: 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.

Identifier	Description
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

105 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").

106 **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

107 Refer to Annex A: Extended Components Definition.

5.3 Functional Requirements

Table 11: 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_RBG_EXT.1	Random Bit Generation
FCS_HTTPS_EXT.1	HTTPS protocol

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	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
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination

Requirement	Title
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 12.*

Table 12: 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.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session	Reason for failure
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSC_EXT.2	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS session	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_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.
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate	Reason for failure of certificate validation
	Any addition, replacement or removal of trust anchors in the TOE's trust store	Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	None	None

Requirement	Auditable Events	Additional Audit Record Contents
FIA_X509_EXT.3	None	None
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Functions	Modification of the behaviour of the transmission of audit data to an external IT entity, the handling of audit data, the audit functionality when Local Audit Storage Space is full.	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	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.

Requirement	Auditable Events	Additional Audit Record Contents
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 12***.

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 the oldest record first*]] 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, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;
- FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [selection: 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: [

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

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

Application note: Changed by TD0581 and TD0580.

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-pass] 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, CTR, GCM] mode and cryptographic key sizes [128 bits, 256 bits] that meet the following: AES as specified in ISO 18033-3, [CBC as specified in ISO 10116, CTR as specified in ISO 10116, GCM as specified in ISO 19772].

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],*
- *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/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_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_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_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with: RFC(s) 4251, 4252, 4253, 4254, [4344, 6668].

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

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

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

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] are the only allowed key exchange methods used for the SSH protocol.

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

Application note: This SFR was altered by TD0631.

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_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_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_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]

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches [the reference identifier per RFC 6125 section 6, IPv4 address in SAN, and no other attribute types].

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

- Not implement any administrator override mechanism].

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

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 TLS Server Protocol Without Mutual Authentication

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:

- 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_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_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_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.

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 perform key establishment for TLS using [Diffie-Hellman parameters with size [2048 bits], ECDHE curves [secp256r1, secp384r1, secp521r1] and no other curves].

FCS_TLSS_EXT.1.4 The TSF shall support [session resumption based on session IDs according to RFC4346 (TLS1.1) or RFC5246 (TLS1.2), session resumption based on session tickets according to RFC 5077].

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 - 3*] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely using a password*.

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

FIA_PMG_EXT.1 Password Management

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

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

d) Minimum password length shall be configurable to between [*8*] and [*32*] *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, SSH public key-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 basic Constraints extension with the CA flag set to TRUE.

- The TSF shall validate the revocation status of the certificate using [a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3]
- 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 manage the cryptographic keys;
    - Ability to configure the cryptographic functionality;
    - Ability to set the time which is used for time-stamps;
    - Ability to manage the TOE's trust store and designate X509v3 certificates as trust anchors
    - Ability to import X509v3 certificates to the TOE's trust store]

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: [

- *BIOS tests*
- *Boot loader image verification*
- *Cryptographic module tests*].

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, [no other capabilities]** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from **disclosure and detection of modification of the channel data**.

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

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

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

108 The TOE security assurance requirements are summarized in Table 13.

Table 13: Assurance Requirements

Assurance Class	Components	Description
Security Target Evaluation	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	Labelling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests	ATE_IND.1	Independent Testing - conformance
Vulnerability Assessment	AVA_VAN.1	Vulnerability Analysis

109 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

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

6.1 Security Audit

6.1.1 FAU_GEN.1

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

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

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

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

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

6.1.2 FAU_GEN.2

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

6.1.3 FAU_STG_EXT.1

114 The TOE is a standalone TOE that stores audit data locally. Logs are written to the FortiManager unit hard disk if the unit contains one. The amount of audit data that can be stored is dependent on the capacity of the device (see Table 5).

115 Local log files can only be deleted via the CLI by an authorized administrator. No editing of log data is permitted.

116 In the evaluated configuration, the TOE is configured to transmit log data to an external FortiAnalyzer platform. Even if an attacker modifies or deletes local data, the TOE will not lose the data because it is also transmitted to an external platform via a secure channel, namely TLS.

117 FortiManager is capable of sending data in real time using the “realtime” option or once-daily using the “upload” option, which uploads the local log at a scheduled time (hh:mm) to the remote FortiAnalyzer unit.

118 If the local storage for audit logs is filled, the TOE overwrites the oldest record first to allow for the saving of a new event.

6.2 Cryptographic Support

6.2.1 FCS_CKM.1

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

a) **RSA 2048.** Used in SSH and TLS.

b) **ECC P-256/P-384/P-521.** Used in TLS.

c) **FFC safe-prime groups.** Diffie-Hellman used in TLS and SSH

6.2.2 FCS_CKM.2

120 The TOE supports the following key establishment schemes:

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

b) **FFC schemes.** Used in SSH key exchange and TLS for supporting DHE ciphersuites. TOE is both a sender and a receiver.

i) Diffie-Hellman Group 14 per RFC 3526 section 3 is supported for FCS_SSHS _EXT.1 and FCS_TLSS_EXT.1

ii) FCS_TLSC_EXT.1/2 is compliant to NIST SP 800-56A Revision 3

121 Table 14 below identifies the scheme being used by each service.

Table 14: Key Agreement Mapping

Scheme	SFR	Service
ECC	FCS_TLSS_EXT.1	Administration
	FCS_TLSC_EXT.1/2	Audit Server
FFC	FCS_SSHS_EXT.1	Administration
	FCS_TLSS_EXT.1	Administration
	FCS_TLSC_EXT.1/2	Audit Server

6.2.3 FCS_CKM.4

122 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 an administrator specified number of times (between 1 and 10) with zeroes. This command is used when a device is reset or taken out of operation. Table 16 shows the origin, storage location and destruction details for cryptographic keys. Unless otherwise stated, the keys are generated by the TOE.

6.2.4 FCS_COP.1/DataEncryption

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

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

6.2.5 FCS_COP.1/SigGen

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

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

b) ECDSA P-256, P-384, P-521

126 The RSA signature verification services are used in for the SSH, TLS and TOE firmware integrity checks.

127 ECDSA is used in TLS.

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

6.2.6 FCS_COP.1/Hash

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

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

a) SSH;

b) TLS;

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

d) Hashing of passwords in non-volatile storage.

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

6.2.7 FCS_COP.1/KeyedHash

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

133 HMAC is implemented in SSH.

134 The characteristics of the HMACs used in the TOE are given in Table 15.

Table 15: 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

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

6.2.8 FCS_RBG_EXT.1

136 The TOE implements an entropy collection system from the hardware-based Fortinet Araneus Alea II Entropy Token. The noise source, which is derived from wide-band radio frequency (RF) white noise, is pooled, and conditioned prior to being used.

137 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.

138 Additional detail is provided the proprietary Entropy Description.

6.2.9 FCS_HTTPS_EXT.1

139 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.

140 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.

6.2.10 FCS_SSHS_EXT.1

141 The TOE implements SSH in compliance with RFCs 4251 through 4254 and 6668.

142 The TOE supports password-based or user public key (ssh-rsa, rsa-sha2-256 and rsa-sha2-512) authentication. The host keys make use of the same algorithms as the user keys.

143 The TOE establishes a user identity by verifying that the non-TOE client uses its private key to send a request to the TOE that only the private key holder can sign and send. The TOE verifies the non-TOE sender using the stored public key.

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

145 The TOE utilizes AES-CTR-128 and AES-CTR-256 for SSH encryption.

146 The TOE provides data integrity for SSH connections via HMAC-SHA1, HMAC-SHA2-256, HMAC-SHA2-384, and HMAC-SHA2-512.

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

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

6.2.11 FCS_TLSC_EXT.1

149 The TOE operates as a TLS client for the trusted channel with the FortiAnalyzer Server.

150 TLS 1.2 is allowed and ciphersuites are restricted to:

a) TLS_DHE_RSA_WITH_AES_128_CBC_SHA

b) TLS_DHE_RSA_WITH_AES_256_CBC_SHA

c) TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

d) TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

e) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

f) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

g) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA

h) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA

i) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

j) TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

k) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

l) TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

151 Ciphersuites are not user-configurable.

152 The reference identifier for the FortiAnalyzer Server is configured by the administrator using the web GUI (IP address) or CLI (IP address or DNS name).

153 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. Otherwise, the reference identifier verification passes and additional verification actions can proceed. The TOE supports wildcards for DNS names in the SAN and CN. IP addresses and DNS names are supported in the SAN. When the SAN is not available, the TOE makes use of the CN, and the connection succeeds. However, CN only supports DNS.

154 The TLS client does not support certificate pinning.

155 The TLS client will transmit the Supported Elliptic Curves extension in the Client Hello message by default with support for the following NIST curves: secp256r1, secp384r1, secp521r1. 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

156 The TOE supports presentation of an X.509v3 client certificate for authentication as required by the FAZ Audit Server.

6.2.13 FCS_TLSS_EXT.1

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

158 The server only allows TLS protocol versions 1.1 and 1.2 (rejecting any other protocol version) and is restricted to the following ciphersuites:

a) TLS_DHE_RSA_WITH_AES_128_CBC_SHA

b) TLS_DHE_RSA_WITH_AES_256_CBC_SHA

c) TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

d) TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

e) TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

f) TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA

g) TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA

h) TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA

i) TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

j) TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

k) TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

l) TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384

159 Ciphersuites are not user-configurable.

160 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. The ECDHE key agreement parameters use secp256r1, secp384r1, secp521r1 and are hardcoded into the server.

161 The TLS server supports session tickets. Session tickets adhere to the structural format provided in section 4 of RFC 5077. Session tickets are encrypted according to the TLS negotiated symmetric encryption algorithm.

162 Session resumption and establishment require session IDs only.

6.3 Identification and Authentication

6.3.1 FIA_PMG_EXT.1

163 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 “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”.

164 The minimum password length is settable by the Administrator and can range from 8 to 32 characters.

6.3.2 FIA_UIA_EXT.1

165 The TOE requires all users to be successfully identified and authenticated. The TOE warning banner is displayed prior to authentication.

166 Administrative access to the TOE is facilitated through several interfaces:

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

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

6.3.3 FIA_UAU_EXT.2

167 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.

168 The TOE provides a local password-based authentication mechanism and also supports SSH public key authentication.

169 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 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.

6.3.4 FIA_UAU.7

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

6.3.5 FIA_AFL.1

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

172 When a user account has sequentially failed authentication the configured number of times the account will be locked for a Security Administrator defined time period.

173 The local console does not implement the lockout mechanism.

6.3.6 FIA_X509_EXT.1/Rev

84 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 CA's, certificate responses and other device-level certificates (such as the web server certificate presented by the TOE TLS web GUI).

85 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;

86 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'.

87 Certificate revocation checking for the above scenarios is performed using a Certificate Revocation List (CRL).

88 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 and 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

174 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.

6.3.7 FIA_X509_EXT.2

85 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.

86 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.

87 As part of the verification process, CRL is used to determine whether the certificate is revoked or not. If the CRL cannot be obtained, then the TOE will choose to accept the certificate in this case.

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

6.3.8 FIA_X509_EXT.3

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

6.4 Security Management

6.4.1 FMT_MOF.1/ManualUpdate

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

6.4.2 FMT_MOF.1/Functions

177 The TOE restricts the ability to configure transmission of audit logs to the FortiAnalyzer to Security Administrators.

6.4.3 FMT_MTD.1/CoreData

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

6.4.4 FMT_SMR.2

179 The TOE maintains the role Security Administrator.

180 Management of TSF data is restricted to Security Administrators.

6.4.5 FMT_MTD.1/CryptoKeys

181 The TOE restricts the ability to modify, delete, generate, import, or otherwise manage SSH keys, TLS, and any configured X.509 certificates or private keys to Security Administrators.

6.4.6 FMT_SMF.1

182 The TOE may be managed via the CLI (console & SSH) or GUI (HTTPS). The TOE provides the following management capabilities:

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

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

e) Ability to configure the authentication failure parameters

f) Ability to manage the cryptographic keys

g) Ability to set the time

h) Ability to manage the TOE’s trust store and designate X509v3 certificates as trust anchors

i) Ability to import X509v3 certificates to the TOE’s trust store

6.5 Protection of the TSF

6.5.1 FPT_SKP_EXT.1

183 Keys are protected as described in Table 16. In all cases, plaintext keys cannot be viewed through an interface designed specifically for that purpose.

Table 16: Keys

Key/CSP	Storage location and method	Usage	Zeroization
Diffie-Hellman Keys	Plaintext in RAM	Key agreement and key establishment	Overwritten with zeroes when no longer needed.
EC Diffie-Hellman Keys	Plaintext in RAM	Key agreement and key establishment	Overwritten with zeroes when no longer needed.
Firmware Update Key	Plaintext in RAM	Verification of firmware integrity when updating to new firmware versions using RSA public key	Overwritten with zeroes when no longer needed.
HTTPS/TLS Server/Host Key	Plaintext in Flash	RSA private key used in the HTTPS/TLS protocols	Overwritten with zeroes when no longer needed.
HTTPS/TLS Session Authentication Key	Plaintext in RAM	HMAC SHA-1, -256 or -384 key used for HTTPS/TLS session authentication	Overwritten with zeroes when no longer needed.
HTTPS/TLS Session Encryption Key	Plaintext in RAM	AES (128-, 256-bit) key used for HTTPS/TLS session encryption	Overwritten with zeroes when no longer needed.
SSH Server/Host Key	Plaintext in Flash	RSA private key used in the SSH protocol (key establishment, 2048- or 3072-bit)	Overwritten with zeroes when no longer needed.
SSH Session Authentication Key	Plaintext in RAM	HMAC SHA-1 or HMAC SHA-256 key used for SSH session authentication	Overwritten with zeroes when no longer needed.
SSH Session Encryption Key	Plaintext in RAM	AES (128-, 256-bit) key used for SSH session encryption	Overwritten with zeroes when no longer needed.
Configuration Encryption Key	Plaintext in Flash	AES 256-bit key used to encrypt CSPs in the boot device and in the backup configuration file (except for crypto officer passwords in the backup configuration file)	Overwritten with zeroes when no longer needed.

6.5.2 FPT_APW_EXT.1

184 Passwords are protected as describe in Table 17. In all cases plaintext passwords cannot be viewed through an interface designed specifically for that purpose.

Table 17: Passwords

Key/Password	Generation/ Algorithm	Storage
Locally stored administrator passwords	User generated	Flash – SHA2-256 hash

6.5.3 FPT_TST_EXT.1

185 The Fortinet family of appliances provides a secure initialization procedure to ensure the integrity of the image and correct cryptographic functioning of the product prior to any information flowing.

186 The product starts from a powered down state and no signals on the wire. The device then powers on and undergoes the following initialization process:

a) Bootstrap and Boot Loader

b) Verification of the kernel, firmware and software images using 2048 bit RSA signature.

c) Loading and Initialization of:

i) Kernel;

ii) Firmware;

iii) Cryptographic known answer tests;

iv) Entropy gathering and DRBG initialization; and

v) Cryptographic module.

187 These tests ensure the correct operation of the TOE, the CPU and BIOS and verify that the correct TOE image is being used. The TOE will not be available if the tests fail. 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 is operating correctly.

6.5.4 FPT_TUD_EXT.1

188 The administrator may query the current version of the TOE via the GUI or CLI.

189 Updates to the TOE are applied in accordance with the following process:

a) The administrator downloads the upgrade image/package from the Fortinet website.

b) Once downloaded, the administrator must transfer the image to the TOE via a trusted path (e.g. the web interface).

c) Upon initiating the update process, the TOE will attempt to verify the integrity and authenticity of the update package. This is achieved via the verification of

a 2048-bit RSA signature that is applied to the package by the Fortinet development team.

d) If the signature cannot be verified, or the integrity of the package cannot be confirmed, the upgrade will fail and an audit log generated accordingly.

e) If the signature is verified correctly and the integrity of the package is confirmed, the upgrade will be applied and the TOE restarted.

6.5.5 FPT_STM_EXT.1

190 The TOE relies on the Security Administrator to manually set the correct time.

191 The TOE makes use of time for the following:

a) Audit record timestamps

b) TOE date/time (to compute expire dates for X.509 certificates)

c) Session timeouts (lockout enforcement)

6.6 TOE Access

6.6.1 FTA_SSL_EXT.1

192 The Security Administrator may configure the TOE to terminate an inactive local interactive session following a specified period of time. This is applicable to the local CLI.

6.6.2 FTA_SSL.3

193 The Security Administrator may configure the TOE to terminate an inactive remote interactive session following a specified period of time. This is applicable to SSH CLI and the HTTPS GUI.

6.6.3 FTA_SSL.4

194 Administrative users may terminate their own sessions at any time.

6.6.4 FTA_TAB.1

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

6.7 Trusted Path/Channels

6.7.1 FTP_ITC.1

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

a) Audit server per FCS_TLSC_EXT.1 and FCS_TLSC_EXT.2

6.7.2 FTP_TRP.1/Admin

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

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

b) **GUI.** Administrative GUI via HTTPS per FCS_HTTPS_EXT.1.

7 Rationale

7.1 Conformance Claim Rationale

198 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

199 All security objectives are drawn directly from the NDcPP.

7.3 Security Requirements Rationale

200 All security requirements are drawn directly from the NDcPP. Table 18 presents a mapping between threats and SFRs as presented in the NDcPP.

Table 18: 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.
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

Identifier	SFR Rationale
	• Requirements for management of updates are defined in FMT_SMF.1 and (for manual updates) in FMT_MOF.1/ManualUpdate, with optional requirements for automatic updates in FMT_MOF.1/AutoUpdate
T.UNDETECTED_ACTIVITY	• Requirements for basic auditing capabilities are specified in FAU_GEN.1 and FAU_GEN.2, with timestamps provided according to FPT_STM_EXT.1 and if applicable, protection of NTP channels in FCS_NTP_EXT.1 • Requirements for protecting audit records stored on the TOE are specified in FAU_STG.1 • Requirements for secure transmission of local audit records to an external IT entity via a secure channel are specified in FAU_STG_EXT.1 • Optional additional requirements for dealing with potential loss of locally stored audit records are specified in FAU_STG_EXT.2/LocSpace, and FAU_STG_EXT.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 • Requirements for secure storage of passwords are set in FPT_APW_EXT.1.
T.SECURITY_FUNCTIONALITY_FAILURE	• Requirements for running self-test(s) are defined in FPT_TST_EXT.1
P.ACCESS_BANNER	• An advisory notice and consent warning message is required to be displayed by FTA_TAB.1

Annex A: Extended Components Definition

201 See appended PDF extract of NDcPP extended components definition.

C. Extended Component Definitions

This appendix contains the definitions for the extended requirements that are used in the cPP, including those used in Appendices A and B.

(Note: formatting conventions for selections and assignments in this Appendix are those in [CC2].)

C.1 Security Audit (FAU)

C.1.1 Security Audit Data Generation (FAU_GEN_EXT)

Family Behaviour

This component defines the requirements for components in a distributed TOE to generate security audit data.

Component levelling

FAU_GEN_EXT.1 Security audit data shall be generated by all components in a distributed TOE

Management: FAU_GEN_EXT.1

The following actions could be considered for the management functions in FMT:

a) The TSF shall have the ability to configure the cryptographic functionality.

Audit: FAU_GEN_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) No audit necessary.

C.1.1.1 FAU_ GEN_EXT.1 Security Audit Data Generation for Distributed TOE Components

FAU_GEN_EXT.1	Security Audit Data Generation
Hierarchical to:	No other components.
Dependencies:	None.

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

C.1.2 Protected Audit Event Storage (FAU_STG_EXT)

Family Behaviour

This component defines the requirements for the TSF to be able to securely transmit audit data between the TOE and an external IT entity.

Component levelling

FAU_STG_EXT.1 Protected audit event storage requires the TSF to use a trusted channel implementing a secure protocol.

FAU_STG_EXT.2 Counting lost audit data requires the TSF to provide information about audit records affected when the audit log becomes full.

FAU_STG_EXT.3 Action in case of possible audit data loss requires the TSF to generate a warning before the audit trail exceeds the local storage capacity.

FAU_STG_EXT.4 Protected Local audit event storage for distributed TOEs requires the TSF to use a trusted channel to protect audit transfer to another TOE component.

FAU_STG_EXT.5 Protected Remote audit event storage for distributed TOEs requires the TSF to use a trusted channel to protect audit transfer to another TOE component.

Management: FAU_STG_EXT.1, FAU_STG_EXT.2, FAU_STG_EXT.3, FAU_STG_EXT.4, FAU_STG_EXT.5

The following actions could be considered for the management functions in FMT:

a) The TSF shall have the ability to configure the cryptographic functionality.

Audit: FAU_STG_EXT.1, FAU_STG_EXT.2, FAU_STG_EXT.3, FAU_STG_EXT.4. FAU_STG_EXT.5

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) No audit necessary.

C.1.2.1 FAU_ STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1	Protected Audit Event Storage

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FTP_ITC.1 Inter-TSF Trusted Channel

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 [selection:

- *The TOE shall consist of a single standalone component that stores audit data locally,*
- *The TOE shall be a distributed TOE that stores audit data on the following TOE components: [assignment: identification of TOE components],*
- *The TOE shall be a distributed TOE with storage of audit data provided externally for the following TOE components: [assignment: list of TOE components that do not store audit data locally and the other TOE components to which they transmit their generated audit data].*

FAU_STG_EXT.1.3 The TSF shall [selection: *drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]*] when the local storage space for audit data is full.

C.1.2.2 FAU_ STG_EXT.2 Counting Lost Audit Data

FAU_STG_EXT.2	Counting Lost Audit Data

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.2.1 The TSF shall provide information about the number of [selection: *dropped, overwritten, [assignment: other information]*] audit records in the case where the local storage has been filled and the TSF takes one of the actions defined in FAU_STG_EXT.1.3.

C.1.2.3 FAU_ STG_EXT.3 Action in Case of Possible Audit Data Loss

FAU_STG_EXT.3	Action in Case of Possible Audit Data Loss

Hierarchical to: No other components.

Dependencies: FAU_GEN.1 Audit data generation
FAU_STG_EXT.1 External Audit Trail Storage

FAU_STG_EXT.3.1/LocSpace The TSF shall *generate a warning to inform the Administrator* before the audit trail *exceeds the local audit trail storage capacity*.

C.1.2.4 FAU_ STG_EXT.4 Protected Local Audit Event Storage for Distributed TOEs

FAU_STG_EXT.4	Protected Audit Event Storage

Hierarchical to: No other components.

Dependencies: FAU_GEN_EXT.1 Security Audit data generation for Distributed TOE Components
[FPT_ITT.1 Intra-TSF Trusted Channel or
FTP_ITC.1 Inter-TSF Trusted Channel]

FAU_STG_EXT.4.1 The TSF of each TOE component which stores security audit data locally shall perform the following actions when the local storage space for audit data is full: *[assignment: table of components and for each component its action chosen according to the following:* [selection: *drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]*]].

C.1.2.5 FAU_ STG_EXT.5 Protected Remote Audit Event Storage for Distributed TOEs

FAU_STG_EXT.5	Protected Audit Event Storage

Hierarchical to: No other components.

Dependencies: FAU_GEN_EXT.1 Security Audit data generation for Distributed TOE Components
[FPT_ITT.1 Intra-TSF Trusted Channel or
FTP_ITC.1 Inter-TSF Trusted Channel]

FAU_STG_EXT.5.1 Each TOE component which does not store security audit data locally shall be able to buffer security audit data locally until it has been transferred to another TOE component that stores or forwards it. All transfer of audit records between TOE components shall use a protected channel according to [selection*: FPT_ITT.1, FTP_ITC.1*].

C.2 Cryptographic Support (FCS)

C.2.1 Random Bit Generation (FCS_RBG_EXT)

C.2.1.1 FCS_RBG_EXT.1 Random Bit Generation

Family Behaviour

Components in this family address the requirements for random bit/number generation. This is a new family defined for the FCS class.

Component levelling

FCS_RBG_EXT.1 Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source.

Management: FCS_RBG_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen

Audit: FCS_RBG_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: failure of the randomization process

FCS_RBG_EXT.1	**Random Bit Generation**
Hierarchical to:	No other components
Dependencies:	No other components

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

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [selection: *[assignment: number of software-based sources] software-based noise source, [assignment: number of platform-based sources] platform-based noise source*] with a minimum of [selection: *128 bits, 192 bits, 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.

C.2.2 Cryptographic Protocols (FCS_DTLSC_EXT, FCS_DTLSS_EXT, FCS_HTTPS_EXT, FCS_IPSEC_EXT, FCS_NTP_EXT, FCS_SSHC_EXT, FCS_SSHS_EXT, FCS_TLSC_EXT, FCS_TLSS_EXT)

C.2.2.1 FCS_DTLSC_EXT DTLS Client Protocol

Family Behaviour

The component in this family addresses the ability for a client to use DTLS to protect data between the client and a server using the DTLS protocol.

Component levelling

FCS_DTLSC_EXT.1 DTLS Client requires that the client side of DTLS be implemented as specified.

FCS_DTLSC_EXT.2 DTLS Client requires that the client side of the DTLS implementation include mutual authentication.

Management: FCS_DTLSC_EXT.1, FCS_DTLSC_EXT.2

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_DTLSC_EXT.1, FCS_DTLSC_EXT.2

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of DTLS session establishment
b) DTLS session establishment
c) DTLS session termination

FCS_DTLSC_EXT.1	**DTLS Client Protocol**
Hierarchical to:	No other components
Dependencies:	FCS_CKM. 1DataEncryption1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment

FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1/SigGen1SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

FCS_DTLSC_EXT.1.1 The TSF shall implement [selection: *DTLS 1.2 (RFC 6347), DTLS 1.0 (RFC 4347)*] supporting the following ciphersuites:

- *[assignment: List of optional ciphersuites and reference to RFC in which each is defined].*

FCS_DTLSC_EXT.1.2 The TSF shall verify that the presented identifier matches [selection: *the reference identifier per RFC 6125 section 6, IPv4 address in CN or SAN, IPv6 address in the CN or SAN, IPv4 address in SAN, IPv6 address in the SAN, the identifier per RFC 5280 Appendix A using [selection: id-at-commonName, id-at-countryName, id-at-dnQualifier, id-at-generationQualifier, id-at-givenName, id-at-initials, id-at-localityName, id-at-name, id-at-organizationalUnitName, id-at-organizationName, id-at-pseudonym, id-at-serialNumber, id-at-stateOrProvinceName, id-at-surname, id-at-title] and no other attribute types*].

FCS_DTLSC_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 [selection:

- *Not implement any administrator override mechanism*
- *require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate*

].

FCS_DTLSC_EXT.1.4 The TSF shall [selection: *not present the Supported Elliptic Curves/Supported Groups Extension, present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups:* [selection: *secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192] and no other curves/groups*] in the Client Hello.

FCS_DTLSC_EXT.2	**DTLS Client Support for Mutual Authentication**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1/DataEncryption Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment

FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FCS_DTLSC_EXT.1 DTLS Client Protocol
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

FCS_DTLSC_EXT.2.1 The TSF shall support mutual authentication using X.509v3 certificates.

FCS_DTLSC_EXT.2.2 The TSF shall [selection: *terminate the DTLS session, silently discard the record*] if a message received contains an invalid MAC.

FCS_DTLSC_EXT.2.3 The TSF shall detect and silently discard replayed messages for:

- DTLS records previously received;
- DTLS records too old to fit in the sliding window.

C.2.2.2 FCS_DTLSS_EXT DTLS Server Protocol

Family Behaviour

The component in this family addresses the ability for a server to use DTLS to protect data between a client and the server using the DTLS protocol.

Component levelling

FCS_DTLSS_EXT.1 DTLS Server requires that the server side of TLS be implemented as specified.

FCS_DTLSS_EXT.2: DTLS Server requires that mutual authentication be included in the DTLS implementation.

Management: FCS_DTLSS_EXT.1, FCS_DTLSS_EXT.2

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_DTLSS_EXT.1, FCS_DTLSS_EXT.2

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of DTLS session establishment.
b) DTLS session establishment
c) DTLS session termination

FCS_DTLSS_EXT.1	**DTLS Server Protocol**

Hierarchical to: No other components

Dependencies:
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1//DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1//SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

FCS_DTLSS_EXT.1.1 The TSF shall implement [selection: *DTLS 1.2 (RFC 6347), DTLS 1.0 (RFC 4347)*] supporting the following ciphersuites:

- *[assignment: List of optional ciphersuites and reference to RFC in which each is defined]*

FCS_DTLSS_EXT.1.2 The TSF shall deny connections from clients requesting *[assignment: list of protocol versions]*.

FCS_DTLSS_EXT.1.3 The TSF shall not proceed with a connection handshake attempt if the DTLS Client fails validation.

FCS_DTLSS_EXT.1.4 The TSF shall perform key establishment for TLS using [selection: *RSA with key size* [selection*: 2048 bits, 3072 bits, 4096 bits], Diffie-Hellman parameters with size* [selection*: 2048 bits, 3072 bits, 4096 bits, 6144 bits, 8192 bits], Diffie-Hellman groups* [selection*: ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, no other groups], ECDHE curves* [selection*: secp256r1, secp384r1, secp521r1] and no other curves*].

FCS_DTLSS_EXT.1.5 The TSF shall [selection: *terminate the DTLS session, silently discard the record*] if a message received contains an invalid MAC.

FCS_DTLSS_EXT.1.6 The TSF shall detect and silently discard replayed messages for:

- DTLS records previously received.
- DTLS Records too old to fit in the sliding window.

FCS_DTLSS_EXT.1.7 The TSF shall support [selection: *no session resumption or session tickets, session resumption based on session IDs according to RFC 4346 (TLS1.1) or RFC 5246 (TLS1.2), session resumption based on session tickets according to RFC 5077*].

FCS_DTLSS_EXT.2	**DTLS Server Support for Mutual Authentication**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification) FCS_COP.1/Hash Cryptographic operation (Hash Algorithm) FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation FCS_DTLSS_EXT.1 DTLS Server Protocol

FCS_DTLSS_EXT.2.1 The TSF shall support mutual authentication of DTLS clients using X.509v3 certificates.

FCS_DTLSS_EXT.2.2 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 [selection:

- *Not implement any administrator override mechanism*
- *require administrator authorization to establish the connection if the TSF fails to* [selection: *match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented client certificate*

].

FCS_DTLSS_EXT.2.3 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 client.

C.2.2.3FCS_HTTPS_EXT.1 HTTPS Protocol

Family Behaviour

Components in this family define the requirements for protecting remote management sessions between the TOE and a Security Administrator. This family describes how HTTPS will be implemented. This is a new family defined for the FCS Class.

Component levelling

FCS_HTTPS_EXT.1 HTTPS requires that HTTPS be implemented according to RFC 2818 and supports TLS.

Management: FCS_HTTPS_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_HTTPS_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) There are no auditable events foreseen.

FCS_HTTPS_EXT.1	**HTTPS Protocol**
Hierarchical to:	No other components
Dependencies:	[FCS_TLSC_EXT.1 TLS Client Protocol, or FCS_TLSS_EXT.1 TLS Server 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 the HTTPS protocol using TLS.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [selection: *not establish the connection, request authorization to establish the connection, [assignment: other action]*] if the peer certificate is deemed invalid.

C.2.2.4 FCS_IPSEC_EXT.1 IPsec Protocol

Family Behaviour

Components in this family address the requirements for protecting communications using IPsec.

Component levelling

FCS_IPSEC_EXT.1 IPsec requires that IPsec be implemented as specified.

Management: FCS_IPSEC_EXT.1

The following actions could be considered for the management functions in FMT:

a) Maintenance of SA lifetime configuration

Audit: FCS_IPSEC_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Decisions to DISCARD, BYPASS, PROTECT network packets processed by the TOE.
b) Failure to establish an IPsec SA
c) IPsec SA establishment
d) IPsec SA termination
e) Negotiation "down" from an IKEv2 to IKEv1 exchange.

FCS_IPSEC_EXT.1	**Internet Protocol Security (IPsec) Communications**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification) FCS_COP.1/Hash Cryptographic operation (Hash Algorithm) FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation

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

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

FCS_IPSEC_EXT.1.3 The TSF shall implement [selection: *tunnel mode, transport mode*].

FCS_IPSEC_EXT.1.4 The TSF shall implement the IPsec protocol ESP as defined by RFC 4303 using the cryptographic algorithms [selection: *AES-CBC-128 (RFC 3602), AES-CBC-192 (RFC 3602), AES-CBC-256 (RFC 3602), AES-GCM-128 (RFC 4106), AES-GCM-192 (RFC 4106), AES-GCM-256 (RFC 4106),*] together with a Secure Hash Algorithm (SHA)-based HMAC [selection: *HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, no HMAC algorithm*].

FCS_IPSEC_EXT.1.5 The TSF shall implement the protocol: [selection:

- *IKEv1, using Main Mode for Phase 1 exchanges, as defined in RFCs 2407, 2408, 2409, RFC 4109,* [selection: *no other RFCs for extended sequence numbers, RFC 4304 for extended sequence numbers*], *and* [selection: *no other RFCs for hash functions, RFC 4868 for hash functions*]*;*

- *IKEv2 as defined in RFCs 5996 [selection: with no support for NAT traversal, with mandatory support for NAT traversal as specified in RFC 5996, section 2.23)*], *and* [selection: *no other RFCs for hash functions, RFC 4868 for hash functions*]].

FCS_IPSEC_EXT.1.6 The TSF shall ensure the encrypted payload in the [selection: *IKEv1, IKEv2*] protocol uses the cryptographic algorithms [selection: *AES-CBC-128, AES_CBC-192 AES-CBC-256 (specified in RFC 3602), AES-GCM-128, AES-GCM-192, AES-GCM-256 (specified in RFC 5282)*].

FCS_IPSEC_EXT.1.7 The TSF shall ensure that [selection:

- *IKEv1 Phase 1 SA lifetimes can be configured by a Security Administrator based on* [selection:
    - *number of bytes;*
    - *length of time, where the time values can be configured within [assignment: integer range including 24*] *hours;*

  ]*;*
- *IKEv2 SA lifetimes can be configured by a Security Administrator based on* [selection:
    - *number of bytes;*
    - *length of time, where the time values can be configured within [assignment: integer range including 24*] *hours*

  ]

].

FCS_IPSEC_EXT.1.8 The TSF shall ensure that [selection:

- *IKEv1 Phase 2 SA lifetimes can be configured by a Security Administrator based on* [selection:
    - *number of bytes;*
    - *length of time, where the time values can be configured within [assignment: integer range including 8] hours;*

];

- *IKEv2 Child SA lifetimes can be configured by a Security Administrator based on* [selection:
  - *number of bytes;*
  - *length of time, where the time values can be configured within [assignment: integer range including 8] hours;*

]

].

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie-Hellman key exchange ("x" in g^x mod p) using the random bit generator specified in FCS_RBG_EXT.1, and having a length of at least *[assignment: (one or more) number(s) of bits that is at least twice the security strength of the negotiated Diffie-Hellman group*] bits.

FCS_IPSEC_EXT.1.10 The TSF shall generate nonces used in [selection: *IKEv1, IKEv2*] exchanges of length [selection:

- *according to the security strength associated with the negotiated Diffie-Hellman group;*
- *at least 128 bits in size and at least half the output size of the negotiated pseudorandom function (PRF) hash*

].

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

- [selection*: 14 (2048-bit MODP), 15 (3072-bit MODP), 16 (4096-bit MODP), 17 (6144-bit MODP), 18 (8192-bit MODP)] according to RFC 3526,*
- [selection*: 19 (256-bit Random ECP), 20 (384-bit Random ECP), 21 (521-bit Random ECP), 24 (2048-bit MODP with 256-bit POS)] according to RFC 5114*.

].

FCS_IPSEC_EXT.1.12 The TSF shall be able to ensure by default that the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: *IKEv1 Phase 1, IKEv2 IKE_SA*] connection is greater than or equal to the strength of the symmetric algorithm (in terms of the number of bits in the key) negotiated to protect the [selection: *IKEv1 Phase 2, IKEv2 CHILD_SA*] connection.

FCS_IPSEC_EXT.1.13 The TSF shall ensure that all IKE protocols perform peer authentication using [selection: *RSA, ECDSA*] that use X.509v3 certificates that conform to RFC 4945 and [selection: *Pre-shared Keys, no other method*].

FCS_IPSEC_EXT.1.14 The TSF shall only establish a trusted channel if the presented identifier in the received certificate matches the configured reference identifier, where the presented and reference identifiers are of the following fields and types: [selection: *SAN: IP address, SAN: Fully Qualified Domain Name (FQDN), SAN: user FQDN, CN: IP address, CN:*

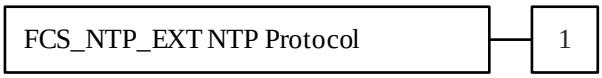
Fully Qualified Domain Name (FQDN), CN: user FQDN, Distinguished Name (DN)] and [selection: *no other reference identifier type, [assignment: other supported reference identifier types*]].

C.2.2.5 FCS_NTP_EXT.1 NTP Protocol

Family Behaviour

The component in this family addresses the ability for a TOE to protect NTP time synchronization traffic.

Component levelling



FCS_NTP_EXT.1 Requires NTP to be implemented as specified

Management: FCS_NTP_EXT.1

The following actions could be considered for the management functions in FMT:

a) Ability to configure NTP

Audit: FCS_NTP_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) No audit requirements are specified.

FCS_NTP_EXT.1	**NTP Protocol**
Hierarchical to:	No other components
Dependencies:	FCS_COP.1 Cryptographic operation [FCS_DTLSC_EXT.1 DTLSC Client Protocol or FCS_IPSEC_EXT.1 IPsec Protocol]

FCS_NTP_EXT.1.1 The TSF shall use only the following NTP version(s) [selection: *NTP v3 (RFC 1305), NTP v4 (RFC 5905)*].

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

- Authentication using [selection: *SHA1, SHA256, SHA384, SHA512, AES-CBC-128, AES-CBC-256*] as the message digest algorithm(s);

- [selection: *IPsec, DTLS*] to provide trusted communication between itself and an NTP time source.

].

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

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

C.2.2.6 FCS_SSHC_EXT.1 SSH Client

Family Behaviour

The component in this family addresses the ability for a client to use SSH to protect data between the client and a server using the SSH protocol.

Component levelling

FCS_SSHC_EXT.1 SSH Client requires that the client side of SSH be implemented as specified.

Management: FCS_SSHC_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_SSHC_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of SSH session establishment
b) SSH session establishment
c) SSH session termination

FCS_SSHC_EXT.1	**SSH Client Protocol**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)

FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_SSHC_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs *4251, 4252, 4253, 4254,* [selection: *4256, 4344, 5647, 5656, 6187, 6668, 8268, 8308 section 3.1, 8332*].

FCS_SSHC_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, [selection: *password-based, no other method*].

FCS_SSHC_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than *[assignment: number of bytes]* bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [assignment: *list of encryption algorithms*].

FCS_SSHC_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [selection: *ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256*] as its public key algorithm(s) and rejects all other public key algorithms

FCS_SSHC_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses *[assignment: list of data integrity MAC algorithms]* as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.7 The TSF shall ensure that *[assignment: list of key exchange methods]* are the only allowed key exchange methods used for the SSH protocol.

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

FCS_SSHC_EXT.1.9 The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key and [selection: *a list of trusted certification authorities, no other methods*] as described in RFC 4251 section 4.1.

C.2.2.7 FCS_SSHS_EXT.1 SSH Server Protocol

Family Behaviour

The component in this family addresses the ability for a server to offer SSH to protect data between a client and the server using the SSH protocol.

Component levelling

FCS_SSHS_EXT.1 SSH Server requires that the server side of SSH be implemented as specified.

Management: FCS_SSHS_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_SSHS_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of SSH session establishment
b) SSH session establishment
c) SSH session termination

FCS_SSHS_EXT.1	**SSH Server Protocol**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification) FCS_COP.1/Hash Cryptographic operation (Hash Algorithm) FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [selection: *4256, 4344, 5647, 5656, 6187, 6668, 8268, 8308* section 3.1, 8332].

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, [selection: *password-based, no other method*].

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than *[assignment: number of bytes]* 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: *[assignment: encryption algorithms]*.

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [selection: *ssh-rsa, rsa-sha2-256, rsa-sha2-512, ecdsa-sha2-nistp256, x509v3-ssh-rsa, ecdsa-sha2-nistp384, ecdsa-sha2-nistp521, x509v3-ecdsa-sha2-nistp256, x509v3-ecdsa-sha2-nistp384, x509v3-ecdsa-sha2-nistp521, x509v3-rsa2048-sha256*] 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 *[assignment: list of MAC algorithms]* as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that *[assignment: list of key exchange methods]* are the only allowed key exchange methods used for the SSH protocol.

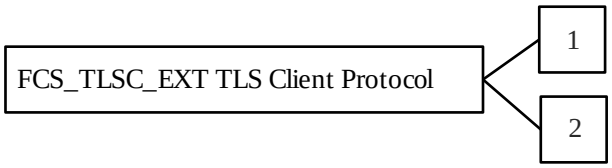
FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections, the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than one gigabyte of data. After any of the thresholds are reached, a rekey needs to be performed.

C.2.2.8 FCS_TLSC_EXT TLS Client Protocol

Family Behaviour

The component in this family addresses the ability for a client to use TLS to protect data between the client and a server using the TLS protocol.

Component levelling



FCS_TLSC_EXT.1 TLS Client requires that the client side of TLS be implemented as specified.

FCS_TLSC_EXT.2 TLS Client requires that the client side of the TLS implementation include mutual authentication.

Management: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of TLS session establishment
b) TLS session establishment
c) TLS session termination

FCS_TLSC_EXT.1	**TLS Client Protocol without Mutual Authentication**

Hierarchical to: No other components

Dependencies: FCS_CKM. 1 Cryptographic Key Generation
FCS_CKM.2 Cryptographic Key Establishment
FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation
FIA_X509_EXT.1 X.509 Certificate Validation
FIA_X509_EXT.2 X.509 Certificate Authentication

FCS_TLSC_EXT.1.1 The TSF shall implement [selection: *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:

- *[assignment: list of optional ciphersuites and reference to RFC in which each is defined]* and no other ciphersuites.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches [selection: *the reference identifier per RFC 6125 section 6, IPv4 address in CN or SAN, IPv6 address in the CN or SAN, IPv4 address in SAN, IPv6 address in the SAN, the identifier per RFC 5280 Appendix A using [selection: id-at-commonName, id-at-countryName, id-at-dnQualifier, id-at-generationQualifier, id-at-givenName, id-at-initials, id-at-localityName, id-at-name, id-at-organizationalUnitName, id-at-organizationName, id-at-pseudonym, id-at-serialNumber, id-at-stateOrProvinceName, id-at-surname, id-at-title] 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 [selection:

- *Not implement any administrator override mechanism*

- *require administrator authorization to establish the connection if the TSF fails to [selection: match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented server certificate*

].

FCS_TLSC_EXT.1.4 The TSF shall [selection: *not present the Supported Elliptic Curves/Supported Groups Extension, present the Supported Elliptic Curves/Supported Groups Extension with the following curves/groups:* [selection*: secp256r1, secp384r1, secp521r1, ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192] and no other curves/groups*] in the Client Hello.

FCS_TLSC_EXT.2	**TLS Client Support for Mutual Authentication**

Hierarchical to: No other components

Dependencies:
- FCS_CKM.1Cryptographic Key Generation
- FCS_CKM.2 Cryptographic Key Establishment
- FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)
- FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)
- FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)
- FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)
- FCS_RBG_EXT.1 Random Bit Generation
- FCS_TLSC_EXT.1 TLS Client Protocol without mutual authentication
- FIA_X509_EXT.1 X.509 Certificate Validation
- FIA_X509_EXT.2 X.509 Certificate Authentication

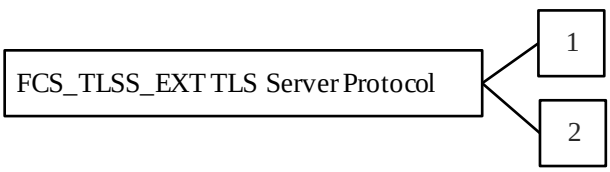
FCS_TLSC_EXT.2.1 The TSF shall support TLS communication with mutual authentication using X.509v3 certificates.

C.2.2.9 FCS_TLSS_EXT TLS Server Protocol

Family Behaviour

The component in this family addresses the ability for a server to use TLS to protect data between a client and the server using the TLS protocol.

Component levelling



FCS_TLSS_EXT.1 TLS Server requires that the server side of TLS be implemented as specified.

FCS_TLSS_EXT.2: TLS Server requires the mutual authentication be included in the TLS implementation.

Management: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of TLS session establishment
b) TLS session establishment
c) TLS session termination

FCS_TLSS_EXT.1	**TLS Server Protocol without Mutual Authentication**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification) FCS_COP.1/Hash Cryptographic operation (Hash Algorithm) FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation FIA_X509_EXT.1 X.509 Certificate Validation FIA_X509_EXT.2 X.509 Certificate Authentication

FCS_TLSS_EXT.1.1 The TSF shall implement [selection: *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:

- *[assignment: list of optional ciphersuites and reference to RFC in which each is defined]* and no other ciphersuites.

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 2.0, SSL 3.0, TLS 1.0 and [selection: *TLS 1.1, TLS 1.2, none*].

FCS_TLSS_EXT.1.3 The TSF shall perform key establishment for TLS using [selection: *RSA with key size* [selection*: 2048 bits, 3072 bits, 4096 bits], Diffie-Hellman parameters with size* [selection*: 2048 bits, 3072 bits, 4096 bits, 6144 bits, 8192 bits], Diffie-Hellman groups*

[selection: *ffdhe2048, ffdhe3072, ffdhe4096, ffdhe6144, ffdhe8192, no other groups], ECDHE curves* [selection: *secp256r1, secp384r1, secp521r1] and no other curves*]].

FCS_TLSS_EXT.1.4 The TSF shall support [selection: *no session resumption or session tickets, session resumption based on session IDs according to* RFC 4346 *(TLS1.1) or* RFC 5246 *(TLS1.2), session resumption based on session tickets according to* RFC 5077].

FCS_TLSS_EXT.2	**TLS Server Support for Mutual Authentication**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FCS_CKM.2 Cryptographic Key Establishment FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption) FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification) FCS_COP.1/Hash Cryptographic operation (Hash Algorithm) FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm) FCS_RBG_EXT.1 Random Bit Generation FCS_TLSS_EXT.1 TLS Server Protocol without mutual authentication FIA_X509_EXT.1 X.509 Certificate Validation FIA_X509_EXT.2 X.509 Certificate Authentication

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

FCS_TLSS_EXT.2.2 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 [selection:

- *Not implement any administrator override mechanism*
- *require administrator authorization to establish the connection if the TSF fails to* [selection: *match the reference identifier, validate certificate path, validate expiration date, determine the revocation status] of the presented client certificate*

].

FCS_TLSS_EXT.2.3 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.

C.3 Identification and Authentication (FIA)

C.3.1 Password Management (FIA_PMG_EXT)

Family Behaviour

The TOE defines the attributes of passwords used by administrative users to ensure that strong passwords and passphrases can be chosen and maintained.

Component levelling

FIA_PMG_EXT.1 Password management requires the TSF to support passwords with varying composition requirements, minimum lengths, maximum lifetime, and similarity constraints.

Management: FIA_PMG_EXT.1

No management functions.

Audit: FIA_PMG_EXT.1

No specific audit requirements.

C.3.1.1 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1	Password Management

Hierarchical to: No other components.

Dependencies: No other components.

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: [selection: *"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [assignment: other characters]*];

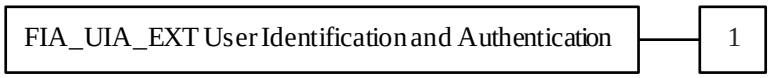
b) Minimum password length shall be configurable to between [*assignment: minimum number of characters supported by the TOE*] and [*assignment: number of characters greater than or equal to 15*] characters.

C.3.2 User Identification and Authentication (FIA_UIA_EXT)

Family Behaviour

The TSF allows certain specified actions before the non-TOE entity goes through the identification and authentication process.

Component levelling



FIA_UIA_EXT.1 User Identification and Authentication requires Administrators (including remote Administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path. It also ensures that every user is identified and authenticated before the TOE performs any mediated functions

Management: FIA_UIA_EXT.1

The following actions could be considered for the management functions in FMT:

a) Ability to configure the list of TOE services available before an entity is identified and authenticated

Audit: FIA_UIA_EXT.N

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) All use of the identification and authentication mechanism
b) Provided user identity, origin of the attempt (e.g. IP address)

C.3.2.1 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1	User Identification and Authentication
Hierarchical to:	No other components.
Dependencies:	FTA_TAB.1 Default TOE Access Banners

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;
- [selection: *no other actions, automated generation of cryptographic keys, [assignment: list of services, actions performed by the TSF in response to non-TOE requests]*].

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.

C.3.3 User authentication (FIA_UAU_EXT)

Family Behaviour

Provides for a locally based administrative user authentication mechanism

Component levelling

FIA_UAU_EXT.2 The password-based authentication mechanism provides administrative users a locally based authentication mechanism.

Management: FIA_UAU_EXT.2

The following actions could be considered for the management functions in FMT:

a) None

Audit: FIA_UAU_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: All use of the authentication mechanism

C.3.3.1 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2	**Password-based Authentication Mechanism**
Hierarchical to:	No other components.
Dependencies:	No other components.

FIA_UAU_EXT.2.1 The TSF shall provide a local [selection: *password-based, SSH public key-based, certificate-based, [assignment: other authentication mechanism(s)]*] authentication mechanism to perform local administrative user authentication.

C.3.4 Authentication using X.509 certificates (FIA_X509_EXT)

Family Behaviour

This family defines the behaviour, management, and use of X.509 certificates for functions to be performed by the TSF. Components in this family require validation of certificates according to a specified set of rules, use of certificates for authentication for protocols and integrity verification, and the generation of certificate requests.

Component levelling

FIA_X509_EXT.1 X509 Certificate Validation, requires the TSF to check and validate certificates in accordance with the RFCs and rules specified in the component.

FIA_X509_EXT.2 X509 Certificate Authentication, requires the TSF to use certificates to authenticate peers in protocols that support certificates, as well as for integrity verification and potentially other functions that require certificates.

FIA_X509_EXT.3 X509 Certificate Requests, requires the TSF to be able to generate Certificate Request Messages and validate responses.

Management: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

The following actions could be considered for the management functions in FMT:

a) Remove imported X.509v3 certificates
b) Approve import and removal of X.509v3 certificates
c) Initiate certificate requests

Audit: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: No specific audit requirements are specified.

C.3.4.1 FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.1	**X.509 Certificate Validation**

Hierarchical to: No other components

Dependencies: FIA_X509_EXT.2 X.509 Certificate Authentication

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

- RFC 5280 certificate validation and certification path validation.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.

- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.

- The TSF shall validate the revocation status of the certificate using [selection: *the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5, no revocation method*]

- The TSF shall validate the extendedKeyUsage field according to the following rules: *[assignment: rules that govern contents of the extendedKeyUsage field that need to be verified]*.

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.

C.3.4.2 FIA_X509_EXT.2 X509 Certificate Authentication

FIA_X509_EXT.2	**X.509 Certificate Authentication**
Hierarchical to:	No other components
Dependencies:	FIA_X509_EXT.1 X.509 Certificate Validation

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for [selection*: DTLS, HTTPS, IPsec, TLS, SSH, [assignment: other protocols], no protocols*], and [selection: *code signing for system software updates [assignment: other uses], 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 [selection: *allow the Administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

C.3.4.3FIA_X509_EXT.3 X.509 Certificate Requests

FIA_X509_EXT.3	**X.509 Certificate Requests**
Hierarchical to:	No other components
Dependencies:	FCS_CKM.1 Cryptographic Key Generation FIA_X509_EXT.1 X.509 Certificate Validation

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 [selection: *device-specific information, Common Name, Organization, Organizational Unit, Country, [assignment: other information]*].

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

C.4 Protection of the TSF (FPT)

C.4.1 Protection of TSF Data (FPT_SKP_EXT)

Family Behaviour

Components in this family address the requirements for managing and protecting TSF data, such as cryptographic keys. This is a new family modelled after the FPT_PTD Class.

Component levelling

FPT_SKP_EXT.1 Protection of TSF Data (for reading all symmetric keys), requires preventing symmetric keys from being read by any user or subject. It is the only component of this family.

Management: FPT_SKP_EXT.1

The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) There are no auditable events foreseen.

C.4.1.1 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)

Hierarchical to: No other components.

Dependencies: No other components.

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

C.4.2 Protection of Administrator Passwords (FPT_APW_EXT)

C.4.2.1 FPT_APW_EXT.1 Protection of Administrator Passwords

Family Behaviour

Components in this family ensure that the TSF will protect plaintext credential data such as passwords from unauthorized disclosure.

Component levelling

FPT_APW_EXT.1 Protection of Administrator passwords requires that the TSF prevent plaintext credential data from being read by any user or subject.

Management: FPT_APW_EXT.1

The following actions could be considered for the management functions in FMT:

a) No management functions.

Audit: FPT_APW_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) No audit necessary.

FPT_APW_EXT.1	**Protection of Administrator Passwords**
Hierarchical to:	No other components
Dependencies:	No other components.

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.

C.4.3 TSF Self-Test (FPT_TST_EXT)

C.4.3.1 FPT_TST_EXT.1 TSF Testing

Family Behaviour

Components in this family address the requirements for self-testing the TSF for selected correct operation.

Component levelling

FPT_TST_EXT.1 TSF Self-Test requires a suite of self-tests to be run during initial start-up in order to demonstrate correct operation of the TSF.

Management: FPT_TST_EXT.1

The following actions could be considered for the management functions in FMT:

a) No management functions.

Audit: FPT_TST_EXT.1

The following actions should be considered for audit if FAU_GEN Security audit data generation is included in the PP/ST:

a) Indication that TSF self-test was completed
b) Failure of self-test

FPT_TST_EXT.1	**TSF Testing**
Hierarchical to:	No other components.
Dependencies:	No other components.

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [selection: *during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]*] to demonstrate the correct operation of the TSF: *[assignment: list of self-tests run by the TSF*].

C.4.4 Trusted Update (FPT_TUD_EXT)

Family Behaviour

Components in this family address the requirements for updating the TOE firmware and/or software.

Component levelling

FPT_TUD_EXT.1 Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.

FPT_TUD_EXT.2 Trusted update based on certificates applies when using certificates as part of trusted update and requires that the update does not install if a certificate is invalid.

Management: FPT_TUD_EXT.1, FPT_TUD_EXT.2

The following actions could be considered for the management functions in FMT:

a) Ability to update the TOE and to verify the updates
b) Ability to update the TOE and to verify the updates using the digital signature capability (FCS_COP.1/SigGen) and [selection: *no other functions, [assignment: other cryptographic functions (or other functions) used to support the update capability]*]
c) Ability to update the TOE, and to verify the updates using [selection: *digital signature, published hash, no other mechanism*] capability prior to installing those updates

Audit: FPT_TUD_EXT.1, FPT_TUD_EXT.2

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Initiation of the update process.
b) Any failure to verify the integrity of the update

C.4.4.1 FPT_TUD_EXT.1 Trusted Update

FPT_TUD_EXT.1	**Trusted Update**
Hierarchical to:	No other components
Dependencies:	FCS_COP.1/SigGen Cryptographic operation (for Cryptographic Signature and Verification), or FCS_COP.1/Hash Cryptographic operation (for cryptographic hashing)

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

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

FPT_TUD_EXT.1.3 The TSF shall provide means to authenticate firmware/software updates to the TOE using a [selection: *X.509 certificate, digital signature, published hash*] prior to installing those updates.

C.4.4.2 FPT_TUD_EXT.2 Trusted Update Based on Certificates

FPT_TUD_EXT.2	Trusted Update Based on Certificates

Hierarchical to: No other components

Dependencies: FPT_TUD_EXT.1

FPT_TUD_EXT.2.1 The TSF shall check the validity of the code signing certificate before installing each update.

FPT_TUD_EXT.2.2 If revocation information is not available for a certificate in the trust chain that is not a trusted certificate designated as a trust anchor, the TSF shall [selection: *not install the update, allow the Administrator to choose whether to accept the certificate in these cases*].

FPT_TUD_EXT.2.3 If the certificate is deemed invalid because the certificate has expired, the TSF shall [selection: *allow the Administrator to choose whether to install the update in these cases, not accept the certificate*].

FPT_TUD_EXT.2.4 If the certificate is deemed invalid for reasons other than expiration or revocation information being unavailable, the TSF shall not install the update.

C.4.5 Time stamps (FPT_STM_EXT)

Family Behaviour

Components in this family extend FPT_STM requirements by describing the source of time used in timestamps.

Component levelling

FPT_STM_EXT.1 Reliable Time Stamps is hierarchic to FPT_STM.1: it requires that the TSF provide reliable time stamps for TSF and identifies the source of the time used in those timestamps.

Management: FPT_STM_EXT.1

The following actions could be considered for the management functions in FMT:

a) Management of the time
b) Administrator setting of the time.

Audit: FTA_SSL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Discontinuous changes to the time.

C.4.5.1FPT_STM_EXT.1 Reliable Time Stamps

FPT_STM_EXT.1	Reliable Time Stamps

Hierarchical to: No other components

Dependencies: No other components.

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 [selection: *allow the Security Administrator to set the time, synchronise time with an NTP server*].

C.5 TOE Access (FTA)

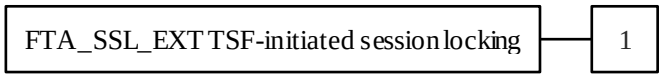
C.5.1 TSF-initiated Session Locking (FTA_SSL_EXT)

Family Behaviour

Components in this family address the requirements for TSF-initiated and user-initiated locking, unlocking, and termination of interactive sessions.

The extended FTA_SSL_EXT family is based on the FTA_SSL family.

Component levelling



FTA_SSL_EXT.1 TSF-initiated session locking, requires system initiated locking of an interactive session after a specified period of inactivity. It is the only component of this family.

Management: FTA_SSL_EXT.1

The following actions could be considered for the management functions in FMT:

c) Specification of the time of user inactivity after which lock-out occurs for an individual user.

Audit: FTA_SSL_EXT.1

The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

b) Any attempts at unlocking an interactive session.

C.5.1.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1	TSF-initiated Session Locking

Hierarchical to: No other components

Dependencies: FIA_UAU.1 Timing of authentication

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

- *lock the session - disable any activity of the Administrator's data access/display devices other than unlocking the session, and requiring that the Administrator re-authenticate to the TSF prior to unlocking the session;*
- *terminate the session*]

after a Security Administrator-specified time period of inactivity.

C.6 Communication (FCO)

C.6.1 Communication Partner Control (FCO_CPC_EXT)

Family Behaviour

This family is used to define high-level constraints on the ways that partner IT entities communicate. For example, there may be constraints on when communication channels can be used, how they are established, and links to SFRs expressing lower-level security properties of the channels.

Component levelling

FCO_CPC_EXT.1 Component Registration Channel Definition, requires the TSF to support a registration channel for joining together components of a distributed TOE, and to ensure that the availability of this channel is under the control of an Administrator. It also requires statement of the type of channel used (allowing specification of further lower-level security requirements by reference to other SFRs).

Management: FCO_CPC_EXT.1

No separate management functions are required. Note that elements of the SFR already specify certain constraints on communication in order to ensure that the process of forming a distributed TOE is a controlled activity.

Audit: FCO_CPC_EXT.1

The following actions should be auditable if FCO_CPC_EXT.1 is included in the PP/ST:

a) Enabling communications between a pair of components as in FCO_CPC_EXT.1.1 (including identities of the endpoints).
b) Disabling communications between a pair of components as in FCO_CPC_EXT.1.3 (including identity of the endpoint that is disabled).

If the required types of channel in FCO_CPC_EXT.1.2 are specified by using other SFRs then the use of the registration channel may be sufficiently covered by the audit requirements on those SFRs: otherwise a separate audit requirement to audit the use of the channel should be identified for FCO_CPC_EXT.1.

C.6.1.1FCO_CPC_EXT.1 Component Registration Channel Definition

FCO_CPC_EXT.1	**Component Registration Channel Definition**

Hierarchical to: No other components.

Dependencies: No other components.

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

FCO_CPC_EXT.1.2 The TSF shall implement a registration process in which components establish and use a communications channel that uses [assignment: *list of different types of channel given in the form of a selection*] for at least [assignment: *type of data for which the channel must be used*].

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