

FORTINET®

FortiProxy 1.0

Security Target

Version 1.0

June 2019

Document prepared by

 **Lightship Security**

www.lightshipsec.com

Document History

Version	Date	Author	Description
1.0	25 June 2019	L Turner	Release for certification

Table of Contents

1	Introduction	4
1.1	Overview	4
1.2	Conformance Claims	4
1.3	Terminology	7
2	TOE Description	8
2.1	Type	8
2.2	Usage	8
2.3	Security Functions	9
2.4	Physical Scope	10
2.5	Logical Scope	10
3	Security Problem Definition	12
3.1	Threats	12
3.2	Assumptions	13
3.3	Organizational Security Policies	14
4	Security Objectives	14
5	Security Requirements	16
5.1	Conventions	16
5.2	Extended Components Definition	16
5.3	Functional Requirements	16
5.4	Assurance Requirements	32
6	TOE Summary Specification	33
6.1	Keys and Passwords	42
7	Rationale	44
7.1	Conformance Claim Rationale	44
7.2	Security Objectives Rationale	44
7.3	Security Requirements Rationale	44
Annex A:	Extended Components Definition	47

List of Tables

Table 1:	Evaluation identifiers	4
Table 2:	NIAP Technical Decisions	4
Table 3:	Terminology	7
Table 4:	CAVP Certificates	9
Table 5:	TOE models	10
Table 6:	Threats	12
Table 7:	Assumptions	13
Table 8:	Organizational Security Policies	14
Table 9:	Security Objectives for the Operational Environment	14
Table 10:	Summary of SFRs	16
Table 11:	Audit Events	18
Table 12:	Assurance Requirements	32
Table 13:	SFR Fulfilment / TOE Summary Specification	33
Table 14:	HMAC Characteristics	35
Table 15:	Keys & Passwords	42
Table 16:	NDcPP SFR Rationale	44

1 Introduction

1.1 Overview

- 1 This Security Target (ST) defines the Fortinet FortiProxy 1.0 Target of Evaluation (TOE) for the purposes of Common Criteria (CC) evaluation.
- 2 Fortinet FortiProxy is a secure web proxy that protects employees against internet-borne attacks by incorporating multiple detection techniques such as web filtering, DNS filtering, data loss prevention, antivirus, intrusion prevention and advanced threat protection. It helps enterprises enforce internet compliance using granular application control. High-performance physical and virtual appliances deploy on-site to serve small, medium and large enterprises.

Table 1: Evaluation identifiers

Target of Evaluation	FortiProxy 1.0 Build: 1.0.7
Security Target	FortiProxy 1.0 Security Target, v1.0

1.2 Conformance Claims

- 3 This ST supports the following conformance claims:
 - a) CC Version 3.1 Release 4
 - b) CC Part 2 extended
 - c) CC Part 3 conformant
 - d) collaborative Protection Profile for Network Devices, v2.0 + Errata 20180314 (NDcPP)
 - e) NIAP Technical Decisions per Table 2

Table 2: NIAP Technical Decisions

TD #	Name	Rationale if n/a
TD0228	NIT Technical Decision for CA certificates - basicConstraints validation	
TD0256	NIT Technical Decision for Handling of TLS connections with and without mutual authentication	
TD0257	NIT Technical Decision for Updating FCS_DTLSC_EXT.x.2/ FCS_TLSC_EXT.x.2 Tests 1-4	
TD0259	NIT Technical Decision for Support for X509 ssh rsa authentication IAW RFC 6187	
TD0281	NIT Technical Decision for Testing both thresholds for SSH rekey	
TD0289	NIT technical decision for FCS_TLSC_EXT.x.1 Test 5e	

TD #	Name	Rationale if n/a
TD0290	NIT technical decision for physical interruption of trusted path/channel	
TD0291	NIT technical decision for DH14 and FCS_CKM.1	
TD0307	Modification of FTP_ITC_EXT.1.1	
TD0321	Protection of NTP communications	TOE does not use NTP
TD0322	NIT Technical Decision for TLS server testing - Empty Certificate Authorities list	FCS_TLSS_EXT.2 not claimed
TD0323	NIT Technical Decision for DTLS server testing - Empty Certificate Authorities list	FCS_DTLSS_EXT.2 not claimed
TD0324	NIT Technical Decision for Correction of section numbers in SD Table 1	
TD0329	IPSEC X.509 Authentication Requirements	
TD0333	NIT Technical Decision for Applicability of FIA_X509_EXT.3	
TD0334	NIT Technical Decision for Testing SSH when password-based authentication is not supported	FCS_SSHC not claimed.
TD0335	NIT Technical Decision for FCS_DTLS Mandatory Cipher Suites	FCS_DTLS not claimed.
TD0336	NIT Technical Decision for Audit requirements for FCS_SSH*_EXT.1.8	
TD0337	NIT Technical Decision for Selections in FCS_SSH*_EXT.1.6	
TD0338	NIT Technical Decision for Access Banner Verification	
TD0339	NIT Technical Decision for Making password-based authentication optional in FCS_SSHS_EXT.1.2	
TD0340	NIT Technical Decision for Handling of the basicConstraints extension in CA and leaf certificates	
TD0341	NIT Technical Decision for TLS wildcard checking	
TD0342	NIT Technical Decision for TLS and DTLS Server Tests	
TD0343	NIT Technical Decision for Updating FCS_IPSEC_EXT.1.14 Tests	FCS_IPSEC not claimed.
TD0394	NIT Technical Decision for Audit of Management Activities related to Cryptographic Keys	

TD #	Name	Rationale if n/a
TD0395	NIT Technical Decision for Different Handling of TLS1.1 and TLS1.2	FCS_TLSS_EXT.2 not claimed
TD0396	NIT Technical Decision for FCS_TLSC_EXT.1.1, Test 2	
TD0397	NIT Technical Decision for Fixing AES-CTR Mode Tests	
TD0398	NIT Technical Decision for FCS_SSH*EXT.1.1 RFCs for AES-CTR	
TD0399	NIT Technical Decision for Manual installation of CRL (FIA_X509_EXT.2)	
TD0400	NIT Technical Decision for FCS_CKM.2 and elliptic curve-based key establishment	
TD0401	NIT Technical Decision for Reliance on external servers to meet SFRs	
TD0402	NIT Technical Decision for RSA-based FCS_CKM.2 Selection	
TD0407	NIT Technical Decision for handling Certification of Cloud Deployments	Not a cloud deployment.
TD0408	NIT Technical Decision for local vs. remote administrator accounts	
TD0409	NIT decision for Applicability of FIA_AFL.1 to key-based SSH authentication	
TD0410	NIT technical decision for Redundant assurance activities associated with FAU_GEN.1	
TD0411	NIT Technical Decision for FCS_SSHC_EXT.1.5, Test 1 - Server and client side seem to be confused	FCS_SSHC not claimed.
TD0412	NIT Technical Decision for FCS_SSHS_EXT.1.5 SFR and AA discrepancy	
TD0423	NIT Technical Decision for Clarification about application of RfI#201726rev2	
TD0425	NIT Technical Decision for Cut-and-paste Error for Guidance AA	

1.3 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

4 The TOE is a network device that provides a secure web gateway, which protects against web attacks with URL filtering, visibility and control of encrypted web traffic, and enforcement of granular web application policies.

2.2 Usage

2.2.1 Deployment

5 As shown in Figure 1, the TOE (enclosed in red) is typically deployed as a gateway between two networks, such as an internal office network and the internet.

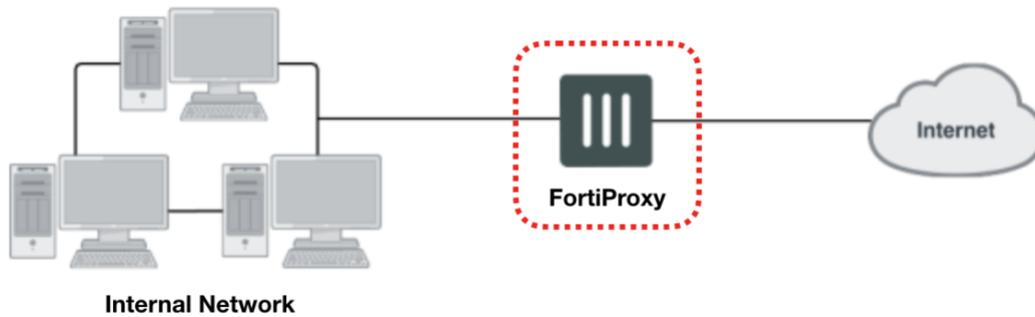


Figure 1: Example TOE Deployment

2.2.2 Interfaces

6 The TOE interfaces are shown in Figure 2.

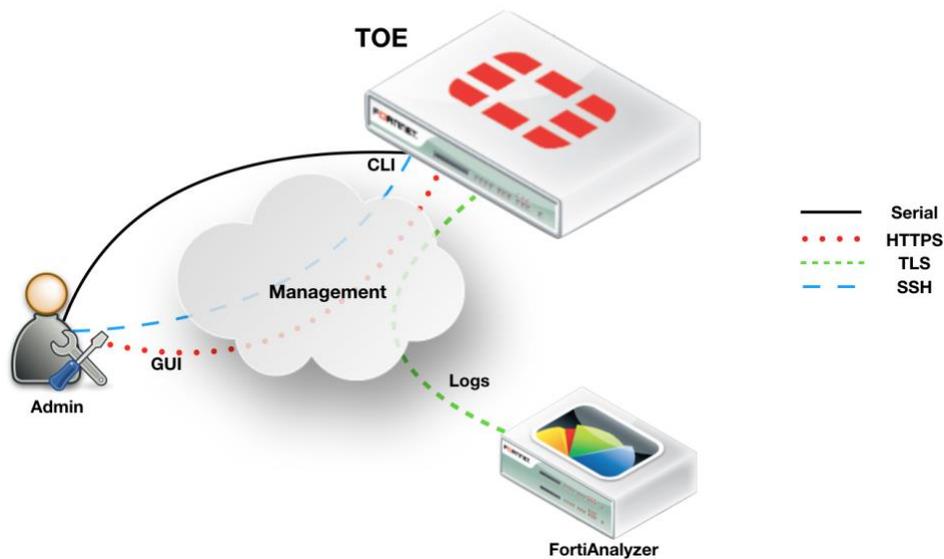


Figure 2: TOE interfaces

- 7 The TOE interfaces are as follows:
- a) **CLI.** Administrative CLI via direct serial connection or SSH.
 - b) **GUI.** Administrative web GUI via HTTPS.
 - c) **Logs.** Forwarding of logs to a remote audit server, which is a Fortinet FortiAnalyzer, via TLS.

2.3 Security Functions

- 8 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 that uses a hardware noise source for entropy. Relevant Cryptographic Algorithm Validation Program (CAVP) certificates are shown in Table 4.

Table 4: CAVP Certificates

Cryptographic Library	Certificate	TOE Services
Fortinet FortiProxy FIPS Cryptographic Library v1.0	C703, C832	Password hashing
Fortinet FortiProxy RBG Cryptographic Library v1.0	C658, C799	DRBG
Fortinet FortiProxy SSL Cryptographic Library v1.0	C655, C702, C787, C806	TLS, SSH, trusted update signature verification
Fortinet CP9 Cryptographic Library v5.6	C813	Hardware acceleration of TLS and SSH

2.4 Physical Scope

9 The physical boundary of the TOE includes the FortiProxy hardware and software which is delivered to the customer via commercial courier. The TOE models in scope are shown in Table 5.

Table 5: TOE models

Model	CPU	RAM	ASIC	CAVP
FPX-400E	Intel i7-4790S	8GB	CP9	C787, C799, C806, C832, C813
FPX-2000E	Intel E5-2680v4	64GB	CP9	C655, C658, C702, C703, C813
FPX-4000E	Dual Intel E5-2680v4	128GB	CP9	

2.4.1 Guidance Documents

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

- a) Fortinet FortiProxy 1.0 FIPS140-2 and Common Criteria Technote, 45-100-528378-20181210
- b) Fortinet FortiProxy 1.0.0 Administration Guide, 45-100-467007-20180411
- c) Fortinet FortiProxy Release Notes Version 1.0.7, Revision 1
- d) Fortinet FortiOS Log Reference Version 5.6.7, 01-565-414447-20181127
- e) Fortinet Hardware Guides:
 - i) Fortinet FortiProxy 400E QuickStart Guide, 45-100-474633-20180413
 - ii) Fortinet FortiProxy 2000E QuickStart Guide, 45-100-477931-20180413
 - iii) Fortinet FortiProxy 4000E QuickStart Guide, 45-100-478166-20180413

11 Guides are available at: <http://docs.fortinet.com/fortiproxy>

2.4.2 Non-TOE Components

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

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

2.5 Logical Scope

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

2.5.1 Functions not included in the TOE Evaluation

- 14 The FortiProxy is capable of a variety of functions and configurations which are not covered by the NDcPP. Only the security functions defined in section 2.3 have been examined as part of this evaluation.
- 15 For the TOE to be in the evaluated configuration, the following functions (which are disabled by default) must not be enabled:
- a) SNMP
 - b) Remote authentication (e.g. RADIUS, LDAP, TACACS+)
 - c) Logging to syslog server
 - d) Logging to FortiCloud

3 Security Problem Definition

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

3.1 Threats

Table 6: Threats

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

Identifier	Description
	the Administrator would have no knowledge that the device has been compromised.
T.SECURITY_ FUNCTIONALITY_ COMPROMISE	Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.
T.PASSWORD_ CRACKING	Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.
T.SECURITY_ FUNCTIONALITY_ FAILURE	A component of the network device may fail during start-up or during operations causing a compromise or failure in the security functionality of the network device, leaving the device susceptible to attackers.

3.2 Assumptions

Table 7: Assumptions

Identifier	Description
A.PHYSICAL_ PROTECTION	The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.
A.LIMITED_ FUNCTIONALITY	The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).
A.NO_THRU_ TRAFFIC_ PROTECTION	A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the NDcPP. It is assumed that this protection will be covered by cPPs for particular types of network devices (e.g., firewall).

Identifier	Description
A.TRUSTED_ADMINISTRATOR	The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.
A.REGULAR_UPDATES	The network device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
A.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the network device are protected by the platform on which they reside.
A.RESIDUAL_INFORMATION	The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

3.3 Organizational Security Policies

Table 8: Organizational Security Policies

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

4 Security Objectives

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

Table 9: Security Objectives for the Operational Environment

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

Identifier	Description
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

5 Security Requirements

5.1 Conventions

18 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").

19 **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

20 Refer to Annex A: Extended Components Definition.

5.3 Functional Requirements

Table 10: Summary of SFRs

Requirement	Title
FAU_GEN.1	Audit Data Generation
FAU_GEN.2	User Identity Association
FAU_STG_EXT.1	Protected Audit Event Storage
FCS_CKM.1	Cryptographic Key Generation
FCS_CKM.2	Cryptographic Key Establishment
FCS_CKM.4	Cryptographic Key Destruction
FCS_COP.1/DataEncryption	Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1/SigGen	Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1/Hash	Cryptographic Operation (Hash Algorithm)
FCS_COP.1/KeyedHash	Cryptographic Operation (Keyed Hash Algorithm)
FCS_HTTPS_EXT.1	HTTPS Protocol
FCS_RBG_EXT.1	Random Bit Generation

Requirement	Title
FCS_SSHS_EXT.1	SSH Server Protocol
FCS_TLSC_EXT.2	TLS Client Protocol with Authentication
FCS_TLSS_EXT.1	TLS Server Protocol
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	Extended: Trusted update
FPT_STM_EXT.1	Reliable Time Stamps
FTA_SSL_EXT.1	TSF-initiated Session Locking
FTA_SSL.3	TSF-initiated Termination
FTA_SSL.4	User-initiated Termination

Requirement	Title
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:*
 - o *Administrative login and logout (name of user account shall be logged if individual user accounts are required for Administrators).*
 - o *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).*
 - o *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).*
 - o *Resetting passwords (name of related user account shall be logged).*
 - o no other actions;
- d) *Specifically defined auditable events listed in ~~Table 2~~ Table 11.*

Table 11: Audit Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_EXT.1	None.	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session.	Reason for failure
FCS_RBG_EXT.1	None.	None.
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.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
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MOF.1/Functions	Modification of the behaviour of the transmission of audit data to an external IT entity, the handling of audit data, the	None.

Requirement	Auditable Events	Additional Audit Record Contents
	audit functionality when Local Audit Storage Space is full.	
FMT_MTD.1/CoreData	All management activities of TSF data.	None.
FMT_MTD.1/CryptoKeys	Management of cryptographic keys.	None.
FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1)	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.

Requirement	Auditable Events	Additional Audit Record Contents
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:
- Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
 - For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, *information specified in column three of ~~Table 2~~ Table 11*.

FAU_GEN.2 User Identity Association

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

FAU_STG_EXT.1 Protected Audit Event Storage

- FAU_STG_EXT.1.1 The TSF shall be able to transmit the generated audit data to an external IT entity using a trusted channel according to FTP_ITC.1.
- FAU_STG_EXT.1.2 The TSF shall be able to store generated audit data on the TOE itself.
- FAU_STG_EXT.1.3 The TSF shall [overwrite previous audit records according to the following rule: [overwrite oldest record first], [no other action] when the local storage space for audit data is full.

5.3.2 Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

- FCS_CKM.1.1 The TSF shall generate **asymmetric** cryptographic keys in accordance with a specified cryptographic key generation algorithm: [
- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;
 - ECC schemes using "NIST curves" [P-256] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;
 - FFC Schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3

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

Application Note: The above SFR is altered by TD0291

FCS_CKM.2 Cryptographic Key Establishment

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

- RSA-based key establishment schemes that meet the following: RSAES-PKCS1-v1_5 as specified in Section 7.2 of RFC 8017, "Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1;
- Elliptic curve-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";
- Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3;

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

Application Note: The above SFR is altered by TD0402.

FCS_CKM.4 Cryptographic Key Destruction

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

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

] that meets the following: *No Standard.*

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

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

FCS_COP.1/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]; ISO/IEC 14888-3, Section 6.4

FCS_COP.1/Hash Cryptographic Operation (Hash Algorithm)

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

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

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

FCS_HTTPS_EXT.1 HTTPS Protocol

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

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

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

FCS_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 hardware based noise source] with a minimum of [256 bits] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs [4251, 4252, 4253, 4254, 6668].

Application Note: TD0398 alters the available selections for the above element.

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

Application Note: The above element is altered by TD0339.

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [262144] bytes in an SSH transport connection are dropped.

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

Application Note: TD0337 altered the available selections for the above element.

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

Application Note: TD0259 altered the available selections for the above element.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [hmac-sha1, hmac-sha2-256, hmac-sha2-512] and [no other MAC algorithms] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

Application Note: TD0337 altered the available selections for the above element.

FCS_SSHS_EXT.1.7 The TSF shall ensure that [diffie-hellman-group14-sha1] and [no other methods] are the only allowed key exchange methods used for the SSH protocol.

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

FCS_TLSC_EXT.2 TLS Client Protocol with authentication

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

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268

- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 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.2.2 The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.

FCS_TLSC_EXT.2.3 The TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [not establish the connection].

FCS_TLSC_EXT.2.4 The TSF shall [present the Supported Elliptic Curves Extension with the following NIST curves: [secp256r1] and no other curves] in the Client Hello.

FCS_TLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

FCS_TLSS_EXT.1 TLS Server Protocol

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

- TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
- TLS_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 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 [generate EC Diffie-Hellman parameters over NIST curves [secp256r1] and no other curves; generate Diffie-Hellman parameters of size 2048 bits and [3072 bits]].

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 - 10] unsuccessful authentication attempts occur related to *Administrators attempting to authenticate remotely*.

FIA_AFL.1.2 When the defined number of unsuccessful authentication attempts has been met, the TSF shall [prevent the offending remote Administrator from successfully authenticating until an Administrator defined time period has elapsed].

FIA_PMG_EXT.1 Password Management

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

- a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!” , “@” , “#” , “\$” , “%” , “^” , “&” , “*” , “(” , “)”];
- b) Minimum password length shall be configurable to [6] and [128]

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;
- [query the TOE version]

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

FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, and [no other 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 certificate path validation **supporting a minimum path length of three certificates**.
- The certificate path must terminate with a trusted CA certificate.
- 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 [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.*

Application Note: The above element is altered by TD0340.

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 [not accept the certificate].

FIA_X509_EXT.3 X.509 Certificate Requests

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

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

5.3.4 Security Management (FMT)

FMT_MOF.1/ManualUpdate Management of security functions behaviour

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

FMT_MOF.1/Functions Management of security functions behaviour

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

FMT_MTD.1/CoreData Management of TSF Data

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

FMT_MTD.1/CryptoKeys Management of TSF data

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

FMT_SMF.1 Specification of Management Functions

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1;*
- [
 - Ability to configure audit behaviour;
 - Ability to set the time which is used for time-stamps;

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 passwords in non-plaintext form.

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

FPT_TST_EXT.1 TSF testing

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [during initial start-up (on power on)] to demonstrate the correct operation of the TSF: [

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

FPT_TUD_EXT.1 Trusted update

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

FPT_TUD_EXT.1.2 The TSF shall provide *Security Administrators* the ability to manually initiate updates to TOE firmware/software and [support automatic checking for updates].

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

FPT_STM_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

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

Table 12: 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

22 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

23

Table 13 describes how the TOE fulfils the SFRs.

Table 13: SFR Fulfilment / TOE Summary Specification

SFR	Fulfilment
FAU_GEN.1	<p>The TOE generates the audit records specified at FAU_GEN.1 containing the following fields:</p> <ul style="list-style-type: none"> • Log number • Date/Time. The time that the log message was created. • Level. The level of the log message. The available logging levels are: <ul style="list-style-type: none"> ○ Alert: Immediate action is required. ○ Critical: Functionality is affected. ○ Error: Functionality is probably affected. ○ Warning: Functionality might be affected. ○ Information: Information about normal events. ○ Debug: Information used for diagnosis or debugging. • User. The user to which the log message relates. User can be a specific user or system. • Message. Detailed log message. <p>The following information is logged as a result of the Security Administrator generating/importing or deleting cryptographic keys:</p> <ul style="list-style-type: none"> • Generate SSH key-pair. Action and key reference. • Generate CSR. Action and key reference. • Import Certificate. Action and key reference. • Import CA Certificate. Action and key reference.
FAU_GEN.2	<p>The TOE includes the user identity in audit events resulting from actions of identified users.</p>
FAU_STG_EXT.1	<p>The Security Administrator can configure the TOE to send logs to a FortiAnalyzer. Log events are cached and sent in near real-time or according to a daily schedule defined by the administrator. Logs are sent via TLS as described by FCS_TLSC_EXT.1.</p> <p>The amount of audit data that may be stored locally is dependent on the available disk space which varies depending on TOE model as shown in Storage column of Table 5.</p> <p>When the local audit data store is full, the TOE will overwrite audit records starting with the oldest audit record.</p> <p>Only authorized administrators may view audit records and no capability to modify the audit records is provided.</p>
FCS_CKM.1	<p>The TOE supports key generation for the following asymmetric schemes:</p>

SFR	Fulfilment
	<ul style="list-style-type: none"> • RSA 2048-bit. Used in SSH and TLS RSA ciphersuites. • ECC P-256. Used in TLS ECC ciphersuites. • Diffie-Hellman Group 14. Diffie-Hellman used in TLS and SSH.
FCS_CKM.2	<p>The TOE supports the following key establishment schemes:</p> <ul style="list-style-type: none"> • RSA schemes. Used in TLS ciphersuites with RSA key exchange. TOE is both sender and receiver. • ECC schemes. Used in TLS ciphersuites with ECDH key exchange. TOE is both sender and receiver. • Diffie-Hellman Group 14. Used in TLS and SSH. The TOE meets RFC 3526 Section 3 by implementing the 2048-bit Modular Exponential (MODP) Group.
FCS_CKM.4	<p>All persistent private keys are encrypted. Keys held in volatile memory are zeroized after use by overwriting the key storage area with zeroes. Keys held in flash memory may be destroyed using a Command Line Interface (CLI) command to overwrite the entire flash memory 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 and passwords. Unless otherwise stated, the keys are generated by the TOE.</p>
FCS_COP.1 /DataEncryption	<p>The TOE provides symmetric encryption and decryption capabilities using 128 and 256 bit AES in CBC and GCM mode. AES is implemented in the following protocols: TLS and SSH (CBC only).</p> <p>The relevant NIST CAVP certificate numbers are listed Table 4.</p>
FCS_COP.1 /SigGen	<p>The TOE provides cryptographic signature generation and verification services using</p> <ul style="list-style-type: none"> • RSA Signature Algorithm with key size of 2048 and greater, • ECDSA Signature Algorithm with NIST curves P-256. <p>These RSA and ECDSA signature verification services are used in the TLS protocols. Additionally, RSA signature verification is used for the SSH protocol (ssh-rsa) and TOE firmware integrity checks.</p> <p>The relevant NIST CAVP certificate numbers are listed in Table 4.</p>
FCS_COP.1 /Hash	<p>The TOE provides cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512.</p> <p>SHS is implemented in the following parts of the TSF:</p> <ul style="list-style-type: none"> • TLS and SSH; • Digital signature verification as part of trusted update validation; and • Hashing of passwords in non-volatile storage. <p>The relevant NIST CAVP certificate numbers are listed in Table 4.</p>

SFR	Fulfilment																				
<p>FCS_COP.1 /KeyedHash</p>	<p>The TOE provides keyed-hashing message authentication services using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512.</p> <p>HMAC is implemented in the following protocols: TLS and SSH.</p> <p>The characteristics of the HMACs used in the TOE are given in Table 14.</p> <p style="text-align: center;">Table 14: HMAC Characteristics</p> <table border="1" data-bbox="448 489 1362 791"> <thead> <tr> <th>Algorithm</th> <th>Block Size</th> <th>Key Size</th> <th>Digest Size</th> </tr> </thead> <tbody> <tr> <td>HMAC-SHA-1</td> <td>512 bits</td> <td>160 bits</td> <td>160 bits</td> </tr> <tr> <td>HMAC-SHA-256</td> <td>512 bits</td> <td>256 bits</td> <td>256 bits</td> </tr> <tr> <td>HMAC-SHA-384</td> <td>1024 bits</td> <td>384 bits</td> <td>384 bits</td> </tr> <tr> <td>HMAC-SHA-512</td> <td>1024 bits</td> <td>512 bits</td> <td>512 bits</td> </tr> </tbody> </table> <p>The relevant NIST CAVP certificate numbers are listed in Table 4.</p>	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
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																		
<p>FCS_HTTPS_EXT.1</p>	<p>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.</p> <p>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.</p>																				
<p>FCS_RBG_EXT.1</p>	<p>The TOE contains a CTR_DRBG that is seeded from the hardware entropy source. Entropy from the noise source is conditioned and used to seed the DRBG with 256 bits of full entropy.</p> <p>Additional detail is provided the proprietary Entropy Description.</p>																				
<p>FCS_SSHS_EXT.1</p>	<p>The TOE implements SSH in compliance with RFCs 4251 through 4254 and 6668.</p> <p>The TOE supports password-based or public key (SSH-RSA) authentication.</p> <p>The TOE examines the size of each received SSH packet. If the packet is greater than 256KB, it is automatically dropped.</p> <p>The TOE utilises AES-CBC-128 and AES-CBC-256 for SSH encryption.</p> <p>The TOE provides data integrity for SSH connections via HMAC-SHA1, HMAC-SHA2-256 and HMAC-SHA2-512.</p> <p>The TOE supports Diffie-Hellman Group 14 SHA-1 (diffie-hellman-group14-sha1) for SSH key exchanges.</p> <p>The TOE will re-key SSH connections after 1 hour of after an aggregate of 1 gig of data has been exchanged (whichever occurs first).</p>																				

SFR	Fulfilment
FCS_TLSC_EXT.2	<p>The TOE operates as a TLS client for the trusted channel with the Audit Server. TLS 1.1 and 1.2 are allowed and ciphersuites are restricted to the following:</p> <ul style="list-style-type: none"> • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_DHE_RSA_WITH_AES_128_CBC_SHA • TLS_DHE_RSA_WITH_AES_256_CBC_SHA • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256 • TLS_RSA_WITH_AES_256_CBC_SHA256 • TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 • TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 <p>Ciphersuites are not user-configurable.</p> <p>The reference identifier for the Audit Server is configured by the administrator using the web GUI. The reference identifiers must be an IP address or hostname.</p> <p>When the TLS client receives an X.509 certificate from the server, the client will compare the reference identifier with the established Subject Alternative Names (SANs) in the certificate. If a SAN is available and does not match the reference identifier, then the verification fails and the channel is terminated. If there are no SANs of the correct type (IP address or DNS name) in the certificate, then the TOE will compare the reference identifier to the Common Name (CN) in the certificate Subject. If there is no CN, then the verification fails and the channel is terminated. If the CN exists and does not match, then the verification fails and the channel is terminated. Otherwise, the reference identifier verification passes and additional verification actions can proceed.</p> <p>The TLS client does not support certificate pinning.</p> <p>The TLS client will transmit the Supported Elliptic Curves extension in the Client Hello message by default with support for the following NIST curves: P256. The non-TOE server can choose to negotiate the elliptic curve from this set for any of the mutually negotiable elliptic curve ciphersuites.</p> <p>The TOE supports presentation of an X.509v3 client certificate for authentication as required by the FAZ Audit Server.</p>

SFR	Fulfilment
FCS_TLSS_EXT.1	<p>The TOE operates as a TLS server for the web GUI trusted path.</p> <p>The server only allows TLS protocol versions 1.1 and 1.2 (rejecting any other protocol version, including SSL 2.0, SSL 3.0 and TLS 1.0 and any other unknown TLS version string supplied) and is restricted to the following ciphersuites by default:</p> <ul style="list-style-type: none"> • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_DHE_RSA_WITH_AES_128_CBC_SHA • TLS_DHE_RSA_WITH_AES_256_CBC_SHA • TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA • TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA • TLS_RSA_WITH_AES_128_CBC_SHA256 • TLS_RSA_WITH_AES_256_CBC_SHA256 • TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 • TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 • TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 • TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 <p>Ciphersuites are not user-configurable.</p> <p>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.</p>
FIA_PMG_EXT.1	<p>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 "!", "@", "#", "\$", "%", "^", "&", "*", "(, ")".</p> <p>The minimum password length is settable by the Administrator and can range from 6 to 128 characters</p>
FIA_UIA_EXT.1 FIA_UAU_EXT.2	<p>The TOE requires all users to be successfully identified and authenticated. The TOE warning banner and TOE version may be viewed prior to authentication.</p> <p>Administrative access to the TOE is facilitated through one of several interfaces,</p> <ul style="list-style-type: none"> • Directly connecting to the TOE appliance via serial for CLI access • Remotely connecting via SSHv2 for CLI access • Remotely connecting via HTTPS for GUI access

SFR	Fulfilment
	<p>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.</p> <p>The TOE provides a local password based authentication mechanism.</p> <p>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.</p>
FIA_UAU.7	<p>For all authentication at the local CLI the TOE displays only "*" characters when the administrative password is entered so that the password is obscured.</p>
FIA_AFL.1	<p>The TOE is capable of tracking authentication failures of remote administrators.</p> <p>When a user account has sequentially failed authentication the configured number of times (default 3), the account will be locked for a Security Administrator defined time period (default 5 minutes).</p> <p>The administrator can configure the maximum number of failed attempts using the web GUI or CLI.</p> <p>The local console does not implement the lockout mechanism.</p>
<p>FIA_X509_EXT.1/Rev</p> <p>FIA_X509_EXT.2</p> <p>FIA_X509_EXT.3</p>	<p>The TOE performs X.509 certificate validation at the following points:</p> <ol style="list-style-type: none"> 1. TOE TLS client validation of server X.509 certificates; 2. When certificates are loaded into the TOE, such as when importing CAs, certificate responses and other device-level certificates (such as the web server certificate presented by the TOE TLS web GUI). <p>In all scenarios, certificates are checked for several validation characteristics:</p> <ul style="list-style-type: none"> • If the certificate 'notAfter' date is in the past, then this is an expired certificate which is considered invalid; • The certificate chain must terminate with a trusted CA certificate; • Server certificates consumed by the TOE TLS client must have a 'serverAuthentication' extendedKeyUsage purpose; <p>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.</p> <p>Certificate revocation checking for the above scenarios is performed using a CRL.</p>

SFR	Fulfilment
	<p>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.</p> <p>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.</p> <p>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:</p> <ul style="list-style-type: none"> • The public key algorithm and parameters are checked • The current date/time is checked against the validity period revocation status is checked • Issuer name of X matches the subject name of X+1 • Name constraints are checked • Policy OIDs are checked • Policy constraints are checked; issuers are ensured to have CA signing bits • Path length is checked • Critical extensions are processed <p>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.</p> <p>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 not accept the certificate in this case.</p> <p>Instructions for configuring the trusted IT entities to supply appropriate X.509 certificates are captured in the guidance documents.</p> <p>For the Certificate Signing Request, a CN is required and may be an IP address, DNS name or email address. SANs are optional and may be email, IP address, URI, DNS name or directory name.</p>
FMT_MOF.1 /Manual Update	The TOE restricts the ability to perform software updates to Security Administrators.
FMT_MOF.1 /Functions	The TOE restricts the ability to modify (enable/disable) transmission of audit records to an external audit server to Security Administrators.
FMT_MTD.1 /CoreData FMT_SMR.2	<p>The TOE implements role-based access control based on pre-defined profiles that are assigned when creating a user. Users are required to login before being provided with access to any administrative functions.</p> <p>All profiles equate to the Security Administrator role within this document</p>

SFR	Fulfilment
	Management of TSF data via the CLI or web GUI is restricted to Security Administrators.
FMT_MTD.1 /CryptoKeys	The TOE restricts the ability to manage SSH, TLS and any configured X.509 private keys to Security Administrators.
FMT_SMF.1	<p>The TOE may be managed via the CLI (console & SSH) or GUI (HTTPS) however the CLI is only intended to be used for initial device configuration and troubleshooting.</p> <p>The specific management capabilities include:</p> <ul style="list-style-type: none"> • 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 • Ability to configure the authentication failure parameters • Ability to configure audit behavior (enable/disable remote logging) • Ability to set the time which is used for time-stamps
FPT_SKP_EXT.1	Keys are protected as described in Table 15.
FPT_APW_EXT.1	Passwords are protected as describe in Table 15.
FPT_TST_EXT.1	<p>At startup, the TOE undergoes the following tests:</p> <ul style="list-style-type: none"> • Firmware integrity test using RSA signatures • Configuration integrity test using HMAC SHA-256 • AES, CBC mode, encrypt known answer test • AES, CBC mode, decrypt known answer test • HMAC SHA-1 known answer test • SHA-1 known answer test (tested as part of HMAC SHA-1 known answer test) • HMAC SHA-256 known answer test • SHA-256 known answer test (tested as part of HMAC SHA-256 known answer test) • HMAC SHA-512 known answer test • SHA-512 known answer test (tested as part of HMAC SHA-512 known answer test) • RSA signature generation known answer test • RSA signature verification known answer test

SFR	Fulfilment
	<ul style="list-style-type: none"> • ECDSA signature generation known answer test • ECDSA signature verification known answer test • DRBG known answer test • Central Processing Unit (CPU) and Memory Basic Input/Output System (BIOS) self-tests – CPU and memory are initialized by exercising a set of known answer tests and the BIOS is compared against a known checksum of the image. The memory is zeroized and then a random pattern is written to and read from the memory. • Boot loader image verification – the boot loader compares the image of the TOE to a known checksum of the image prior to booting. <p>These tests ensure the correct operation of the cryptographic functionality of the TOE, the CPU and BIOS and verify that the correct TOE image is being used. The cryptographic functionality will not be available if the tests fail, and any operation of the TOE supported by this functionality will not be available. If the CPU, or BIOS tests fail, the device will not complete the boot up operation. If the boot loader image verification fails, the boot up operation will fail. When the device completes the boot up operation, this is evidence that the self-tests have passed, and that the TOE, and the cryptographic functions are operating correctly.</p> <p>The cryptographic module executes the following conditional tests when the related service is invoked:</p> <ul style="list-style-type: none"> • Continuous NDRNG test • Continuous DRBG test • RSA pairwise consistency test • ECDSA pairwise consistency test • Configuration integrity test using HMAC SHA-256 • Firmware load test using RSA signatures <p>Self-tests can also be initiated on demand using the CLI command 'execute fips kat all' (to initiate all self-tests) or 'execute fips kat <test>' (to initiate a specific self-test). If a self-test fails, the device enters error mode and halts system operation. All data output and cryptographic services are inhibited when in the error state. Continued operation indicates that the tests have passed, and the TOE is operating correctly.</p>
FPT_TUD_EXT.1	<p>The current firmware version may be queried using 'status' in the CLI, or by viewing 'System Information' using the web UI.</p> <p>The administrator may download a firmware update from https://support.fortinet.com using a network workstation. The firmware may then be updated using the web UI by selecting 'System Information' and selecting 'Update'. The process to upload the firmware image file includes verification of the digital signature using the Fortinet firmware update key (RSA 2048), which is held in an encoded format in the current firmware image. If the digital signature on the firmware image cannot be verified, an error message appears. Once uploaded and verified, the new image is automatically installed; then the TOE reboots and the firmware is automatically activated.</p>

SFR	Fulfilment
	The TOE will automatically check for updates by default. This behavior cannot be disabled. When an update is available, a notification is presented to the administrator who may choose to download and install the update.
FPT_STM_EXT.1	<p>The TOE incorporates an internal clock that is used to maintain date and time. The Security Administrator sets the date and time during initial TOE configuration and may change the time during operation.</p> <p>The TOE makes used of time for the following:</p> <ul style="list-style-type: none"> • Audit record timestamps • Session timeouts (lockout enforcement) • Certificate validation
FTA_SSL_EXT.1 FTA_SSL.3 FTA_SSL.4	The Security Administrator may configure the TOE to terminate an inactive session following a specified period of time. The timeout value is set to five minutes by default, and applies to both local and remote CLI and web UI sessions. Administrative users may terminate their own sessions at any time.
FTA_TAB.1	The TOE displays an administrator configurable message to users prior to login at the CLI (console & SSH) and web GUI. The same message is displayed at all interfaces.
FTP_ITC.1	<p>The TOE supports secure communication with the following IT entities:</p> <ul style="list-style-type: none"> • Audit server per FCS_TLSC_EXT.2 (TOE is client) <p>This connection is protected via TLS as described in the related SFRs.</p>
FTP_TRP.1/ Admin	<p>The TOE provides the following trusted paths for remote administration:</p> <ul style="list-style-type: none"> • CLI over SSH per FCS_SSHS_EXT.1 • Web GUI over HTTPS per FCS_HTTPS_EXT.1.1

6.1 Keys and Passwords

24 Table 15 describes the keys and passwords used in the TOE. In all cases, keys and passwords cannot be viewed through an interface designed specifically for that purpose.

Table 15: Keys & Passwords

Key/Password	Generation/Algorithm	Storage	Zeroization
TLS Private Key	RSA (2048 bits) or ECDSA (P-256, P-384, P-521)	Flash - plaintext	Overwritten with zeroes by erase-disk command.

Key/Password	Generation/Algorithm	Storage	Zeroization
TLS Public Key	RSA (2048 bits) or ECDSA (P-256, P-384, P-521)	Flash - plaintext	n/a – public key
DH Keys used for TLS	DH (2048 bits)	RAM - plaintext	Overwritten with zeroes upon termination of the session or reboot of the appliance
AES key used for TLS	AES-128 AES-256	RAM - plaintext	Overwritten with zeroes upon termination of the session or reboot of the appliance
Firmware Update Key (Public Key)	Preconfigured RSA (2048 bits)	Flash - plaintext	n/a – public key
Locally stored administrator passwords	User generated	Flash - SHA-256 hash	Overwritten with zeroes by erase-disk command.
SSH Private Key (host key)	RSA (2048 bits)	Flash - plaintext	Overwritten with zeroes by erase-disk command.
SSH Public Key	RSA (2048 bits)	Flash - plaintext	n/a – public key
SSH Session Key	AES-128 AES-256	RAM - plaintext	The keys (including re-keyed keys) are overwritten with zeroes when no longer required or reboot of the appliance

7 Rationale

7.1 Conformance Claim Rationale

25 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

26 All security objectives are drawn directly from the NDcPP.

7.3 Security Requirements Rationale

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

Table 16: NDcPP SFR Rationale

Identifier	SFR Rationale
T.UNAUTHORIZED_ADMINISTRATOR_ACCESS	<ul style="list-style-type: none"> 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
	<ul style="list-style-type: none"> • (Protection of the Administrator credentials is separately addressed by T.PASSWORD_CRACKING).
T.WEAK_CRYPTOGRAPHY	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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	<ul style="list-style-type: none"> • 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
	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Requirements for basic auditing capabilities are specified in FAU_GEN.1 and FAU_GEN.2, with timestamps provided according to FPT_STM_EXT.1 Requirements for protecting audit records stored on the TOE are specified in FAU_STG_EXT.1 Requirements for secure transmission of local audit records to an external IT entity via a secure channel are specified in FAU_STG_EXT.1 Optional additional requirements for dealing with potential loss of locally stored audit records are specified in FAU_STG_EXT.2/LocSpace, and FAU_STG.3/LocSpace If (optionally) configuration of the audit functionality is provided by the TOE then this is specified in FMT_SMF.1, and confining this functionality to Security Administrators is required by FMT_MOF.1/Functions.
T.SECURITY_FUNCTIONALITY_COMPROMISE	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> 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	<ul style="list-style-type: none"> Requirements for running self-test(s) are defined in FPT_TST_EXT.1 Optional use of certificates to support self-test(s) is defined in FPT_TST_EXT.2 (with support for the use of certificates in FIA_X509_EXT.1, FIA_X509_EXT.2, and FIA_X509_EXT.3),

Annex A: Extended Components Definition

28 This annex reproduces the NDcPP Appendix C extended components definition.

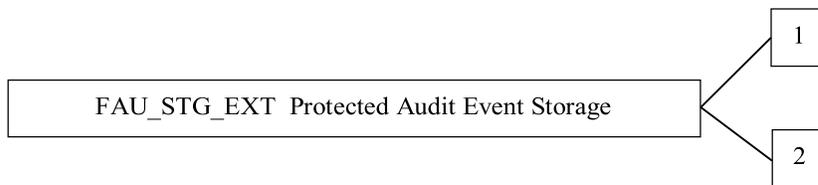
C.1 Security Audit (FAU)

C.1.1 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.

Management: FAU_STG_EXT.1, FAU_STG_EXT.2

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

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_STG_EXT.1 Protected Audit EventStorage

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.

Application Note 127

For selecting the option of transmission of generated audit data to an external IT entity the TOE relies on a non-TOE audit server for storage and review of audit records. The storage of these audit records and the ability to allow the Administrator to review these audit records is provided by the operational environment in that case. Since the external audit server is not part of the TOE, there are no requirements on it except the capabilities for ITC transport for audit data. No requirements are placed upon the format or underlying protocol of the audit data being transferred. The TOE must be capable of being configured to transfer audit data to an external IT entity without Administrator intervention. Manual transfer would not meet the requirements. Transmission could be done in real-time or periodically. If the transmission is not done in real-time then the TSS describes what event stimulates the transmission to be made and what range of frequencies the TOE supports for making transfers of audit data to the audit server; the TSS also suggests typical acceptable frequencies for the transfer.

For distributed TOEs each component must be able to export audit data across a protected channel external (FTP_ITC.1) or intercomponent (FPT_ITT.1 or FTP_ITC.1) as appropriate. At least one component of the TOE must be able to export audit records via FTP_ITC.1 such that all TOE audit records can be exported to an external IT entity.

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

FAU_STG_EXT.1.3 The TSF shall [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.

Application Note 128

The external log server might be used as alternative storage space in case the local storage space is full. The “other action” could in this case be defined as “send the new audit data to an external IT entity”.

For distributed TOEs each component must provide some amount of local storage to ensure that audit records are preserved in case of network connectivity issues. The behaviour when local storage is exhausted must be described for each component.

C.1.1.2 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.

Application Note 129

This option should be chosen if the TOE supports this functionality.

In case the local storage for audit records is cleared by the Administrator, the counters associated with the selection in the SFR should be reset to their initial value (most likely to 0). The guidance documentation should contain a warning for the Administrator about the loss of audit data when he clears the local storage for audit records.

For distributed TOEs each component that implements counting of lost audit data has to provide a mechanism for Administrator access to, and management of, this information.

If FAU_STG_EXT.2 is added to the ST, the ST has to make clear any situations in which lost audit data is not counted.

C.2 Cryptographic Support (FCS)

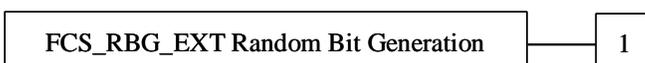
C.2.1 Random Bit Generation (FCS_RBG_EXT)

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 hardware-based sources] hardware-based noise source*] with 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.

Application Note 130

For the first selection in FCS_RBG_EXT.1.2, the ST selects at least one of the types of noise sources. If the TOE contains multiple noise sources of the same type, the ST author fills the assignment with the appropriate number for each type of source (e.g., 2 software-based noise sources, 1 hardware-based noise source). The documentation

and tests required in the Evaluation Activity for this element necessarily describes each source indicated in the ST.

ISO/IEC 18031:2011 contains three different methods of generating random numbers; each of these, in turn, depends on underlying cryptographic primitives (hash functions/ciphers). The ST author will select the function used, and include the specific underlying cryptographic primitives used in the requirement. While any of the identified hash functions (SHA-1, SHA- 224, SHA-256, SHA-384, SHA-512) are allowed for Hash_DRBG or HMAC_DRBG, only AES- based implementations for CTR_DRBG are allowed.

C.2.2 Cryptographic Protocols (FCS_DTLSC_EXT, FCS_DTLSS_EXT, FCS_HTTPS_EXT, FCS_IPSEC_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.1 DataEncryption1 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.1.1 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*]

Application Note 131

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the ciphersuites that are supported. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. TLS_RSA_WITH_AES_128_CBC_SHA is not mandatory for ND cPP v2.0 compliance; however, it is required if claiming compliance with RFC 6347.

These requirements will be revisited as new DTLS versions are standardized by the IETF. In a future version of this cPP DTLS v1.2 will be required for all TOEs.

FCS_DTLSC_EXT.1.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125 section 6.

Application Note 132

The rules for verification of identity are described in Section 6 of RFC 6125. The reference identifier is established by the Administrator (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the DTLS server's certificate.

FCS_DTLSC_EXT.1.3 The TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [selection: *not establish the connection, request authorization to establish the connection, [assignment: other action]*].

Application Note 133

If DTLS is selected in FTP_ITC then validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/Rev. If DTLS is selected in FPT_ITT, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/ITT.

FCS_DTLSC_EXT.1.4 The TSF shall [selection: *not present the Supported Elliptic Curves Extension, present the Supported Elliptic Curves Extension with the following NIST curves: [selection: secp256r1, secp384r1, secp521r1] and no other curves*] in the Client Hello.

Application Note 134

If ciphersuites with elliptic curves were selected in FCS_DTLSC_EXT.1.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS_DTLSC_EXT.1.1, then "not present the Supported Elliptic Curves Extension" should be selected.

This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1/SigGen and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.

Hierarchical to: FCS_DTLSC_EXT.1 DTLS Client Protocol

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.2.1 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*].

Application Note 135

The ST author should select the ciphersuites that are supported. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. TLS_RSA_WITH_AES_128_CBC_SHA is not mandatory for ND cPP v2.0 compliance; however, it is required if claiming compliance with RFC 6347. These requirements will be revisited as new DTLS versions are standardized by the IETF. In a future version of this cPP DTLS v1.2 will be required for all TOEs.

FCS_DTLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125 section 6.

Application Note 136

The rules for verification of identity are described in Section 6 of RFC 6125. The reference identifier is established by the Administrator (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the DTLS server's certificate.

FCS_DTLSC_EXT.2.3 The TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [selection: *not establish the connection, request authorization to establish the connection, [assignment: other action]*].

Application Note 137

If DTLS is selected in FTP_ITC then validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

Certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/Rev. If DTLS is selected in FPT_ITT, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/ITT.

FCS_DTLSC_EXT.2.4 The TSF shall [selection: *not present the Supported Elliptic Curves Extension, present the Supported Elliptic Curves Extension with the following NIST curves: [selection: *secp256r1, secp384r1, secp521r1*] and no other curves*] in the Client Hello].

Application Note 138

If ciphersuites with elliptic curves were selected in FCS_DTLSC_EXT.2.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS_DTLS_EXT.2.1, then “not present the Supported Elliptic Curves Extension” should be selected.

This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1/SigGen and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.

FCS_DTLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

Application Note 139

The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include the client must be capable of presenting a certificate to a DTLS server for DTLS mutual authentication.

FCS_DTLSC_EXT.2.6 The TSF shall [selection: *terminate the DTLS session, silently discard the record*] if a message received contains an invalid MAC.

Application Note 140

The Message Authentication Code (MAC) is keyed hash function specified in FCS_COP.1/KeyedHash. The MAC is negotiated during DTLS handshake phase and is used to protect integrity of messages received from the sender during DTLS data exchange. If MAC verification fails, the session must be terminated or the record must be silently discarded.

FCS_DTLSC_EXT.2.7 The TSF shall detect and silently discard replayed messages for:

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

Application Note 141

Replay Detection is described in section 4.1.2.6 of DTLS 1.2 (RFC 6347) and section 4.1.2.5 of DTLS 1.0 (RFC 4347). For each received record, the receiver verifies the record contains a sequence number is within the sliding receive window and does not duplicate the sequence number of any other record received during the session.

"Silently Discard" means the TOE discards the packet responding.

C.2.1.2 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 the 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

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*]

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the ciphersuites that are supported. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. TLS_RSA_WITH_AES_128_CBC_SHA is not mandatory for ND cPP v2.0 compliance; however, it is required if claiming compliance with RFC 6347. These requirements will be revisited as new DTLS versions are standardized by the IETF. In a future version of this cPP DTLS v1.2 will be required for all TOEs.

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

Application Note 142

This version of the cPP does not require the TOE to deny DTLS v1.0. In a future version of this cPP DTLS v1.0 will be required to be denied for all TOEs.

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

Application Note 143

The process to validate the IP address of a DTLS client is specified in section 4.2.1 of RFC 6347 (DTLS 1.2) and RFC 4347 (DTLS 1.0). The TOE validates the DTLS client during Connection Establishment (Handshaking) and prior to the TSF sending a Server Hello message. After receiving a ClientHello, the DTLS Server sends a HelloVerifyRequest along with a cookie. The cookie is a signed message using the keyed hash function specified in FCS_COP.1 /KeyedHash. The DTLS Client then sends another ClientHello with the cookie attached. If the DTLS server successfully verifies the signed cookie, the Client is not using a spoofed IP address.

FCS_DTLSS_EXT.1.4 The TSF shall [selection: *perform RSA key establishment with key size* [selection: 2048 bits, 3072 bits, 4096 bits]; generate EC Diffie-Hellman parameters over NIST curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; generate Diffie- Hellman parameters of size [selection: 2048, bits, 3072 bits]].

Application Note 144

If the ST lists a DHE or ECDHE ciphersuite in FCS_DTLSS_EXT.1.1, the ST must include the Diffie-Hellman or NIST curves selection in the requirement. FMT_SMF.1 requires the configuration of the key agreement parameters in order to establish the security strength of the DTLS connection.

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.

Application Note 145

The Message Authentication Code (MAC) is keyed hash function specified in FCS_COP.1/KeyedHash. The MAC is negotiated during DTLS handshake phase and is used to protect integrity of messages received from the sender during DTLS data

exchange. If MAC verification fails, the session must be terminated or the record must be silently discarded.

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.

Application Note 146

Replay Detection is described in section 4.1.2.6 of DTLS 1.2 (RFC 6347) and section 4.1.2.5 of DTLS 1.0 (RFC 4347). For each received record, the receiver verifies the record contains a sequence number is within the sliding receive window and does not duplicate the sequence number of any other record received during the session.

"Silently Discard" means the TOE discards the packet without responding.

FCS_DTLSS_EXT.2	DTLS Server Protocol with mutual authentication
------------------------	--

Hierarchical to:	FCS_DTLSS_EXT.1 DTLS Server Protocol
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.2.1 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*].

Application Note 147

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the ciphersuites that are supported. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment.

TLS_RSA_WITH_AES_128_CBC_SHA is not mandatory for ND cPP v2.0 compliance; however, it is required if claiming compliance with RFC 6347.

These requirements will be revisited as new DTLS versions are standardized by the IETF. In a future version of this cPP DTLS v1.2 will be required for all TOEs.

FCS_DTLSS_EXT.2.2 The TSF shall deny connections from clients requesting [assignment: *list of protocol versions*].

Application Note 148

This version of the cPP does not require the TOE to deny DTLS v1.0. In a future version of this cPP DTLS v1.0 will be required to be denied for all TOEs.

FCS_DTLSS_EXT.2.3 The TSF shall not proceed with a connection handshake attempt if the DTLS Client fails validation.

Application Note 149

The process to validate the IP address of a DTLS client is specified in section 4.2.1 of RFC 6347 (DTLS 1.2) and RFC 4347 (DTLS 1.0). The TOE validates the DTLS client during Connection Establishment (Handshaking) and prior to the TSF sending a Server Hello message. After receiving a ClientHello, the DTLS Server sends a HelloVerifyRequest along with a cookie. The cookie is a signed message using the keyed hash function specified in FCS_COP.1/KeyedHash. The DTLS Client then sends another ClientHello with the cookie attached. If the DTLS server successfully verifies the signed cookie, the Client is not using a spoofed IP address.

FCS_DTLSS_EXT.2.4 The TSF shall [selection: *perform RSA key establishment with key size* [selection: 2048 bits, 3072 bits, 4096 bits]; generate EC Diffie-Hellman parameters over NIST curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; generate Diffie- Hellman parameters of size [selection: 2048, bits, 3072 bits]].

Application Note 150

If the ST lists a DHE or ECDHE ciphersuite in FCS_DTLSS_EXT.2.1, the ST must include the Diffie-Hellman or NIST curves selection in the requirement. FMT_SMF.1 requires the configuration of the key agreement parameters in order to establish the security strength of the DTLS connection.

FCS_DTLSS_EXT.2.5 The TSF shall [selection: *terminate the DTLS session, silently discard the record*] if a message received contains an invalid MAC.

Application Note 151

The Message Authentication Code (MAC) is negotiated during the DTLS handshake phase and is used to protect integrity of messages received from the sender during DTLS data exchange. If MAC verification fails, the session must be terminated or the record must be silently discarded.

FCS_DTLSS_EXT.2.6 The TSF shall detect and silently discard replayed messages for:

- DTLS records that have previously been received.
- DTLS records too old to fit in the sliding window.

Application Note 152

Replay Detection is described in section 4.1.2.6 of DTLS 1.2 (RFC 6347) and section 4.1.2.5 of DTLS 1.0 (RFC 4347). For each received record, the receiver verifies the record contains a sequence number is within the sliding receive window and does not duplicate the sequence number of any other record received during the session.

"Silently Discard" means the TOE discards the packet without responding.

FCS_DTLSS_EXT.2.7 The TSF shall support mutual authentication of DTLS clients using X.509v3 certificates.

FCS_DTLSS_EXT.2.8 The TSF shall not establish a trusted channel if the client certificate is invalid. If the client certificate is deemed invalid, then the TSF shall [selection: *not establish the connection, request authorization to establish the connection, [assignment: other action]*].

Application Note 153

The use of X.509v3 certificates for DTLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include support for client-side certificates for DTLS mutual authentication.

If DTLS is selected in FTP_ITC then validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280. Certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/Rev. If DTLS is selected in FPT_ITT, then certificate validity is tested in accordance with testing performed for FIA_X509_EXT.1/ITT.

FCS_DTLSS_EXT.2.9 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.

Application Note 154

The client identifier may be in the Subject field or the Subject Alternative Name extension of the certificate. The expected identifier may either be configured, may be compared to the Domain Name, IP address, username, or email address used by the peer, or may be passed to a directory server for comparison. Matching should be performed by a bit-wise comparison.

C.2.1.3FCS_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.

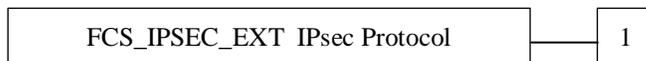
FCS_HTTPS_EXT.1.3 The TSF shall [selection: *not establish the connection, request authorization to establish the connection, [assignment: other action]*] if the peer certificate is deemed invalid.

C.2.1.4FCS_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:	<p>FCS_CKM.1 Cryptographic Key Generation</p> <p>FCS_CKM.2 Cryptographic Key Establishment</p> <p>FCS_COP.1/DataEncryption Cryptographic operation (AES Data encryption/decryption)</p> <p>FCS_COP.1/SigGen Cryptographic operation (Signature Generation and Verification)</p> <p>FCS_COP.1/Hash Cryptographic operation (Hash Algorithm)</p> <p>FCS_COP.1/KeyedHash Cryptographic operation (Keyed Hash Algorithm)</p> <p>FCS_RBG_EXT.1 Random Bit Generation</p>

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

Application Note 155

RFC 4301 calls for an IPsec implementation to protect IP traffic through the use of a Security Policy Database (SPD). The SPD is used to define how IP packets are to be handled: PROTECT the packet (e.g., encrypt the packet), BYPASS the IPsec services (e.g., no encryption), or DISCARD the packet (e.g., drop the packet). The SPD can be implemented in various ways, including router access control lists, firewall rulesets, a “traditional” SPD, etc. Regardless of the implementation details, there is a notion of a “rule” that a packet is “matched” against and a resulting action that takes place.

While there must be a means to order the rules, a general approach to ordering is not mandated, as long as the SPD can distinguish the IP packets and apply the rules accordingly. There may be multiple SPDs (one for each network interface), but this is not required.

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, AES-CBC-192, AES-CBC-256 (specified in RFC 3602), no other algorithm*] together with a Secure Hash Algorithm (SHA)-based HMAC [selection: *HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512, no other algorithm*] and [selection: *AES-*

GCM-128, AES-GCM-192, AES-GCM-256 (specified in RFC 4106), no other 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)*].

Application Note 156

AES-GCM-128, AES-GCM-192 and AES-GCM-256 may only be selected if IKEv2 is also selected, as there is no RFC defining AES-GCM for IKEv1.

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*

]

].

Application Note 157

The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either volume- based lifetimes or time-based lifetimes (or a combination). This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE). Hardcoded limits do not meet this requirement. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the guidance documentation generated for AGD_OPE.

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;*
-]

].

Application Note 158

The ST author chooses either the IKEv1 requirements or IKEv2 requirements (or both, depending on the selection in FCS_IPSEC_EXT.1.5). The ST author chooses either volume- based lifetimes or time-based lifetimes (or a combination). This requirement must be accomplished by providing Security Administrator-configurable lifetimes (with appropriate instructions in documents mandated by AGD_OPE). Hardcoded limits do not meet this requirement. In general, instructions for setting the parameters of the implementation, including lifetime of the SAs, should be included in the guidance documentation generated for AGD_OPE.

FCS_IPSEC_EXT.1.9 The TSF shall generate the secret value x used in the IKE Diffie- Hellman key exchange (“ x ” in $g^x \text{ mod } p$) 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.

Application Note 159

For DH groups 19 and 20, the "x" value is the point multiplier for the generator point G. Since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1.9 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management – Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment for this element. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.

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

- *[assignment: 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*
-].

Application Note 160

The ST author must select the second option for nonce lengths if IKEv2 is also selected (as this is mandated in RFC 5996). The ST author may select either option for IKEv1.

For the first option for nonce lengths, since the implementation may allow different Diffie-Hellman groups to be negotiated for use in forming the SAs, the assignment in FCS_IPSEC_EXT.1.10 may contain multiple values. For each DH group supported, the ST author consults Table 2 in NIST SP 800-57 "Recommendation for Key Management –Part 1: General" to determine the security strength ("bits of security") associated with the DH group. Each unique value is then used to fill in the assignment for this element. For example, suppose the implementation supports DH group 14 (2048-bit MODP) and group 20 (ECDH using NIST curve P-384). From Table 2, the bits of security value for group 14 is 112, and for group 20 it is 192.

Because nonces may be exchanged before the DH group is negotiated, the nonce used should be large enough to support all TOE-chosen proposals in the exchange.

FCS_IPSEC_EXT.1.11 The TSF shall ensure that IKE protocols implement DH Group(s) [selection: 14 (2048-bit MODP), 19 (256-bit Random ECP), 24 (2048-bit MODP with 256-bit POS), 20 (384-bit Random ECP)] and [selection: 5 (1536-bit MODP), no other group].

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.

Application Note 161

The ST author chooses either or both of the IKE selections based on what is implemented by the TOE. While it is acceptable for this capability to be configurable, the default configuration in the evaluated configuration (either "out of the box" or by configuration guidance in the AGD documentation) must enable this functionality.

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

C.2.1.5 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.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_SSHC_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs [selection: 4251, 4252, 4253, 4254, 5647, 5656, 6187, 6668, *no other RFCs*].

Application Note 162

The ST author selects which of the RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are “REQUIRED”. This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as “REQUIRED” but not listed in the later elements of this component are implemented is out of scope of the evaluation activity for this requirement.

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.

Application Note 163

RFC 4253 provides for the acceptance of “large packets” with the caveat that the packets should be of “reasonable length” or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining “reasonable length” for the TOE.

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, 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 no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

Application Note 164

This SFR defines two thresholds - one for the maximum time span the same session keys can be used and the other one for the maximum amount of data that can be transmitted using the same session keys. Both thresholds need to be implemented and a rekey needs to be performed on whichever threshold is reached first. For the maximum transmitted data threshold, the total incoming and outgoing data needs to be counted. The rekey applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.

It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR.

For any configurable threshold related to this requirement the guidance documentation needs to specify how the threshold can be configured. The allowed values must either be specified in the guidance documentation and must be lower or equal to the thresholds specified in this SFR or the TOE must not accept values beyond the thresholds specified in this SFR.

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 [selection: *public key, a list of trusted certification authorities, no other methods*] as described in RFC 4251 section 4.1.

Application Note 165

The list of trusted certification authorities can only be selected if x509v3-ecdsa-sha2-nistp256 or x509v3-ecdsa-sha2-nistp384 are specified in FCS_SSHC_EXT.1.5.

C.2.1.6 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.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_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs [selection: 4251, 4252, 4253, 4254, 5647, 5656, 6187, 6668, *no other RFCs*].

Application Note 166

The ST author selects which of the RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are “REQUIRED”. This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as “REQUIRED” but not listed in the later elements of this component are implemented is out of scope of the evaluation activity for this requirement.

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

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

Application Note 167

RFC 4253 provides for the acceptance of “large packets” with the caveat that the packets should be of “reasonable length” or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining “reasonable length” for the TOE.

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, 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 no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

Application Note 168

This SFR defines two thresholds - one for the maximum time span the same session keys can be used and the other one for the maximum amount of data that can be transmitted using the same session keys. Both thresholds need to be implemented and a rekey needs to be performed on whichever threshold is reached first. For the maximum transmitted data threshold, the total incoming and outgoing data needs to be counted. The rekey applies to all session keys (encryption, integrity protection) for incoming and outgoing traffic.

It is acceptable for a TOE to implement lower thresholds than the maximum values defined in the SFR.

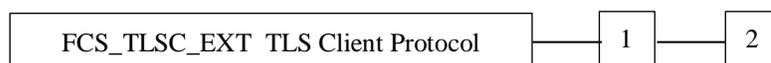
For any configurable threshold related to this requirement the guidance documentation needs to specify how the threshold can be configured. The allowed values must either be specified in the guidance documentation and must be lower or equal to the thresholds specified in this SFR or the TOE must not accept values beyond the thresholds specified in this SFR.

C.2.1.7 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
-----------------------	----------------------------

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_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 TSF shall support the following ciphersuites: [selection:

- *Mandatory Ciphersuites:*
 - *[assignment: list of mandatory ciphersuites and reference to RFC in*

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

Application Note 169

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS_RSA_WITH_AES_128_CBC_SHA is not mandatory for ND cPP v2.0, but is required to ensure compliance with RFC 5246.

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

Application Note 170

The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

FCS_TLSC_EXT.1.3 The TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall *[selection: not establish the connection, request authorization to establish the connection, [assignment: other action]]*.

Application Note 171

Validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

FCS_TLSC_EXT.1.4 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: *[assignment: list of supported curves including an option for "none"]*.

Application Note 172

If ciphersuites with elliptic curves were selected in FCS_TLSC_EXT.1.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS_TLS_EXT.1.1, then “none” should be selected.

This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1/SigGen and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.

FCS_TLSC_EXT.2	TLS Client Protocol with Authentication
-----------------------	--

Hierarchical to: FCS_TLSC_EXT.1 TLS Client Protocol

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

- *Mandatory Ciphersuites:*
 - *[assignment: list of mandatory ciphersuites and reference to RFC in which each is defined]*
- *[selection: Optional Ciphersuites:*
 - *[assignment: list of optional ciphersuites and reference to RFC in which each is defined];*
 - *no other ciphersuite]]].*

Application Note 173

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS_RSA_WITH_AES_128_CBC_SHA is not mandatory for ND cPP/FW cPP v2.0, but is required to ensure compliance with RFC 5246.

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125.

Application Note 174

The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

FCS_TLSC_EXT.2.3 The TSF shall only establish a trusted channel if the server certificate is valid. If the server certificate is deemed invalid, then the TSF shall [selection: *not establish the connection, request authorization to establish the connection, [assignment: other action]*].

Application Note 175

Validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

FCS_TLSC_EXT.2.4 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [assignment: *list of supported curves including an option for "none"*].

Application Note 176

If ciphersuites with elliptic curves were selected in FCS_TLSC_EXT.1.1, a selection of one or more curves is required. If no ciphersuites with elliptic curves were selected in FCS_TLS_EXT.1.1, then "none" should be selected.

This requirement limits the elliptic curves allowed for authentication and key agreement to the NIST curves from FCS_COP.1/SigGen and FCS_CKM.1 and FCS_CKM.2. This extension is required for clients supporting Elliptic Curve ciphersuites.

FCS_TLSC_EXT.2.5 The TSF shall support mutual authentication using X.509v3 certificates.

Application Note 177

The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.

C.2.1.8 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
-----------------------	----------------------------

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

- *Mandatory Ciphersuites:*
 - *[assignment: list of mandatory ciphersuites and reference to RFC in which each is defined]*
 - *[selection: Optional Ciphersuites:*
 - *[assignment: list of optional ciphersuites and reference to RFC in which each is defined];*
- no other ciphersuite]]].*

Application Note 178

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that TLS_RSA_WITH_AES_128_CBC_SHA is not mandatory for ND cPP v2.0, but is required in order to ensure compliance with RFC 5246.

FCS_TLSS_EXT.1.2 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*].

Application Note 179

All SSL versions and TLS v1.0 are denied. Any TLS versions not selected in FCS_TLSS_EXT.1.1 should be selected here. (If “none” is the selection for this element then the ST author may omit the words “and none”.)

FCS_TLSS_EXT.1.3 The TSF shall generate key establishment parameters using RSA with key size [selection: *2048 bits*, *3072 bits*, *4096 bits*, *no other size*] and

[selection: [assignment: List of elliptic curves]; [assignment: list of Diffie-Hellman parameter sizes]].

Application Note 180

The assignments will be filled in based on the assignments performed in FCS_TLSS_EXT.1.1.

FCS_TLSS_EXT.2	TLS Server Protocol with mutual authentication
-----------------------	---

Hierarchical to: FCS_TLSS_EXT.1 TLS Server Protocol

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

- *Mandatory Ciphersuites:*
 - [assignment: list of mandatory ciphersuites and reference to RFC in which each is defined]
 - [selection: *Optional Ciphersuites:*
 - [assignment: list of optional ciphersuites and reference to RFC in which each is defined];
- no other ciphersuite]]].*

Application Note 181

The ciphersuites to be tested in the evaluated configuration are limited by this requirement. Note that *TLS_RSA_WITH_AES_128_CBC_SHA* is not mandatory for ND CPP v2.0, but is required in order to ensure compliance with RFC 5246.

FCS_TLSS_EXT.2.2 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*].

Application Note 182

All SSL versions and TLS v1.0 are denied. Any TLS versions not selected in FCS_TLSS_EXT.1.1 should be selected here. (If “none” is the selection for this element then the ST author may omit the words “and none”.)

FCS_TLSS_EXT.2.3 The TSF shall generate key establishment parameters using RSA with key size [selection: *2048 bits, 3072 bits, 4096 bits, no other size*] and [selection: *[assignment: list of elliptic curves]; [assignment: list of Diffie-Hellman parameter sizes]*].

Application Note 183

The assignments will be filled in based on the assignments performed in FCS_TLSS_EXT.2.1.

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

Application Note 184

The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include support for client-side certificates for TLS mutual authentication.

FCS_TLSS_EXT.2.5 The TSF shall not establish a trusted channel if the client certificate is invalid. If the client certificate is deemed invalid, then the TSF shall [selection: *not establish the connection, request authorization to establish the connection, [assignment: other action]*].

Application Note 185

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

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

Application Note 186

This requirement only applies to those TOEs performing mutually-authenticated TLS (FCS_TLSS_EXT.2.4). The peer identifier may be in the Subject field or the Subject Alternative Name extension of the certificate. The expected identifier may either be configured, may be compared to the Domain Name, IP address, username, or email address used by the peer, or may be passed to a directory server for comparison.

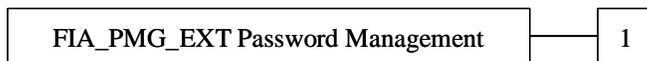
C.3 Identification and Authentication (FIA)

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:

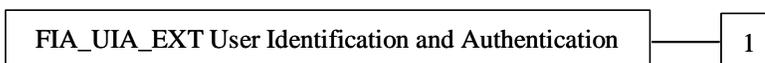
- a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [selection: “!”, “@”, “#”, “\$”, “%”, “^”, “&”, “*”, “(”, “)”, [assignment: other characters]];
- b) Minimum password length shall be settable by the Security Administrator, and support passwords of 15 characters or greater.

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.3.2.2 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*, [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.

Application Note 187

This requirement applies to users (Administrators and external IT entities) of services available from the TOE directly, and not services available by connecting through the TOE. While it should be the case that few or no services are available to external entities prior to identification and authentication, if there are some available (perhaps ICMP echo) these should be listed in the assignment statement; otherwise “no other actions” should be selected.

Authentication can be password-based through the local console or through a protocol that supports passwords (such as SSH), or be certificate based (such as DTLS, SSH, TLS).

For communications with external IT entities (e.g., an audit server or NTP server, for instance), such connections must be performed in accordance with FTP_ITC.1, whose protocols perform identification and authentication. This means that such communications (e.g., establishing the IPsec connection to the authentication server) would not have to be specified in the assignment, since establishing the connection “counts” as initiating the identification and authentication process.

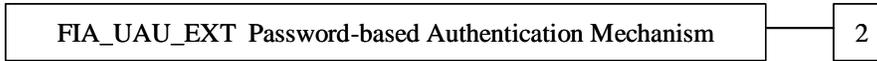
According to the application note for FMT_SMR.2, for distributed TOEs at least one TOE component has to support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2 but not necessarily all TOE components. In case not all TOE components support this way of authentication for Security Administrators the TSS shall describe how Security Administrators are authenticated and identified.

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 password-based authentication mechanism, [selection: *[assignment: other authentication mechanism(s)], none*] to perform administrative user authentication.

Application Note 188

The assignment should be used to identify any additional local authentication mechanisms supported. Local authentication mechanisms are defined as those that occur through the local console; remote administrative sessions (and their associated authentication mechanisms) are specified in FTP_TRP.1/Admin.

According to the application note for FMT_SMR.2, for distributed TOEs at least one TOE component has to support the authentication of Security Administrators according to FIA_UIA_EXT.1 and FIA_UAU_EXT.2 but not necessarily all TOE components. In case not all TOE components support this way of authentication for Security Administrators the TSS shall describe how Security Administrators are authenticated and identified.

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.3 X.509 Certificate Requests

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

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [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*].

Application Note 189

FIA_X509_EXT.1.1 lists the rules for validating certificates. The ST author selects whether revocation status is verified using OCSP or CRLs. If the TOE is distributed and X.509 based authentication is being used to authenticate the protocol selected in FPT_ITT.1, certificate revocation checking is optional. It is optional because there are additional requirements surrounding the enabling and disabling of the FPT_ITT

channel defined in FCO_CPC_EXT.1. If revocation is not supported the ST author selects no revocation method. The ST author fills in the assignment with rules that may apply to other requirements in the ST. For instance, if a protocol such as TLS that uses certificates is specified in the ST, then certain values for the extendedKeyUsage field (e.g., “Server Authentication Purpose”) could be specified.

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.

Application Note 190

This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.

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.3 X.509 Certificate Requests

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, code signing for integrity verification, [assignment: other uses], no additional uses*].

Application Note 191

If the TOE specifies the implementation of communications protocols that perform peer authentication using certificates, the ST author either selects or assigns the protocols that are specified; otherwise, they select “no protocols”. Protocols that do not use X.509 based peer authentication include SSH, where ssh-rsa, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and/or ecdsa-sha2-nistp521 are selected. The TOE may also use certificates for other purposes; the second selection and assignment are used to specify these cases.

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*].

Application Note 192

Often a connection must be established to check the revocation status of a certificate - either to download a CRL or to perform a lookup using OCSP. The selection is used to describe the behaviour in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate valid according to all other rules in FIA_X509_EXT.1, the behaviour indicated in the selection determines the validity. The TOE must not accept the certificate if it fails any of the other validation rules in FIA_X509_EXT.1. If the Administrator-configured option is selected by the ST Author, the ST Author also selects the corresponding function in FMT_SMF.1.

If the TOE is distributed and FIA_X509_EXT.1/ITT is selected, then certificate revocation checking is optional. This is due to additional authorization actions being performed in the enabling and disabling of the intra-TOE trusted channel as defined in FCO_CPC_EXT.1. In this case, a connection is not required to determine certificate validity and this SFR is trivially satisfied..

C.3.4.3 FIA_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.2 X.509
 Certificate Authentication

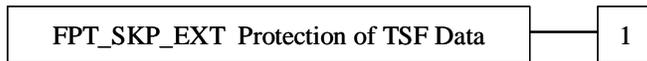
FIA_X509_EXT.3.1 The TSF shall generate a Certificate Request Message as specified by RFC 2986 and be able to provide the following information in the request: public key and [selection: *device-specific information, Common Name, Organization, Organizational Unit, Country, [assignment: other information]*].

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.

Application Note 193

The intent of this requirement is for the device to protect keys, key material, and authentication credentials from unauthorized disclosure. This data should only be

accessed for the purposes of their assigned security functionality, and there is no need for them to be displayed/accessed at any other time. This requirement does not prevent the device from providing indication that these exist, are in use, or are still valid. It does, however, restrict the reading of the values outright.

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 passwords in non-plaintext form.

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

C.4.3 TSF Self-Test (FPT_TST_EXT)

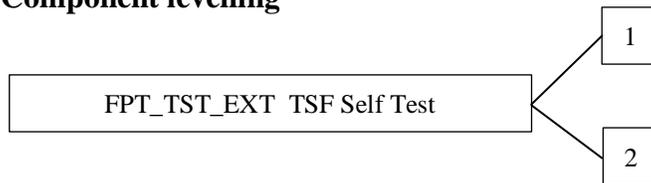
C.4.3.1 FPT_TST_EXT.1

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.

FPT_TST_EXT.2 Self-tests based on certificates applies when using certificates as part of self- test, and requires that the self-test fails if a certificate is invalid.

Management: FPT_TST_EXT.1, FPT_TST_EXT.2

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

- a) No management functions.

Audit: FPT_TST_EXT.1, FPT_TST_EXT.2

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*].

Application Note 194

It is expected that self-tests are carried out during initial start-up (on power on). Other options should only be used if the developer can justify why they are not carried out during initial start-up. It is expected that at least self-tests for verification of the integrity of the firmware and software as well as for the correct operation of cryptographic functions necessary to fulfil the SFRs will be performed. If not all self-tests are performed during start-up multiple iterations of this SFR are used with the appropriate options selected. In future versions of this cPP the suite of self-tests will be required to contain at least mechanisms for measured boot including self-tests of the components which perform the measurement.

For distributed TOEs all TOE components have to perform self-tests. This does not necessarily mean that each TOE component has to carry out the same self-tests: the ST describes the applicability of the selection (i.e. when self-tests are run) and the final assignment (i.e. which self-tests are carried out) to each TOE component.

Application Note 195

If certificates are used by the self-test mechanism (e.g. for verification of signatures for integrity verification), certificates are validated in accordance with FIA_X509_EXT.1 and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TST_EXT.2 must be included in the ST.

FPT_TST_EXT.2	Self-tests based on certificates
----------------------	---

Hierarchical to: No other components.

Dependencies: No other components.

FPT_TST_EXT.2.1 The TSF shall fail self-testing if a certificate is used for self-tests and the corresponding certificate is deemed invalid.

Application Note 196

Certificates may optionally be used for self-tests (FPT_TST_EXT.1.1). This element must be included in the ST if certificates are used for self-tests. If “code signing for integrity verification” is selected in FIA_X509_EXT.2.1, FPT_TST_EXT.2 must be included in the ST.

Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with FIA_X509_EXT.1.

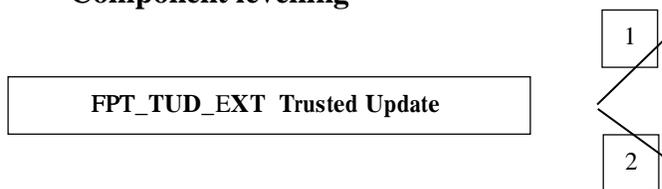
C.4.4 Trusted Update

(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

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

FPT_TUD_EXT.1	Trusted Update
----------------------	-----------------------

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)

FPT_TUD_EXT.1.1 The TSF shall provide [assignment: *Administrators*] the ability to query the currently executing version of the TOE firmware/software as well as the most recently installed version of the TOE firmware/software.

Application Note 197

The version currently running (being executed) may not be the version most recently installed. For instance, maybe the update was installed but the system requires a reboot before this update will run. Therefore, it needs to be clear that the query should indicate both the most recently executed version as well as the most recently installed update.

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*].

Application Note 198

The selection in FPT_TUD_EXT.1.2 distinguishes the support of automatic checking for updates and support of automatic updates. The first option refers to a TOE that checks whether a new update is available, communicates this to the Administrator (e.g. through a message during an Administrator session, through log files) but requires some action by the Administrator to actually perform the update. The second option refers to a TOE that checks for updates and automatically installs them upon availability.

The TSS explains what actions are involved in the TOE support when using the “support automatic checking for updates” or “support automatic updates” selections.

When published hash values (see FPT_TUD_EXT.1.3) are used to protect the trusted update mechanism, the TOE must not automatically download the update file(s) together with the hash value (either integrated in the update file(s) or separately) and automatically install the update without any active authorization by the Security Administrator, even when the calculated hash value matches the published hash value. When using published hash values to protect the trusted update mechanism, the option “support of automatic updates” must not be used (automated checking for updates is permitted, though). The TOE may automatically download the update file(s) themselves but not to the hash value. For the published hash approach, it is intended that a Security Administrator is always required to give active authorisation for installation of an update (as described in more detail under FPT_TUD_EXT.1.3) below. Due to this, the type of update mechanism is regarded as “manually initiated update”, even if the update file(s) may be downloaded automatically. A fully automated approach (without Security Administrator intervention) can only be used when “digital signature mechanism” is selected in FPT_TUD_EXT.1.3 below.

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

Application Note 199

The digital signature mechanism referenced in the selection of FPT_TUD_EXT.1.3 is one of the algorithms specified in FCS_COP.1/SigGen. The published hash referenced in FPT_TUD_EXT.1.3 is generated by one of the functions specified in FCS_COP.1/Hash. The ST author should choose the mechanism implemented by the TOE; it is acceptable to implement both mechanisms.

When published hash values are used to secure the trusted update mechanism, an active authorization of the update process by the Security Administrator is always required. The secure transmission of an authentic hash value from the developer to the Security Administrator is one of the key factors to protect the trusted update mechanism when using published hashes and the guidance documentation needs to describe how this

transfer has to be performed. For the verification of the trusted hash value by the Security Administrator different use cases are possible. The Security Administrator could obtain the published hash value as well as the update file(s) and perform the verification outside the TOE while the hashing of the update file(s) could be done by the TOE or by other means. Authentication as Security Administrator and initiation of the trusted update would in this case be regarded as “active authorization” of the trusted update. Alternatively, the Administrator could provide the TOE with the published hash value together with the update file(s) and the hashing and hash comparison is performed by the TOE. In case of successful hash verification, the TOE can perform the update without any additional step by the Security Administrator. Authentication as Security Administrator and sending the hash value to the TOE is regarded as “active authorization” of the trusted update (in case of successful hash verification), because the Administrator is expected to load the hash value only to the TOE when intending to perform the update. As long as the transfer of the hash value to the TOE is performed by the Security Administrator, loading of the update file(s) can be performed by the Security Administrator or can be automatically downloaded by the TOE from a repository.

If the digital signature mechanism is selected, the verification of the signature shall be performed by the TOE itself. For the published hash option, the verification can be done by the TOE itself as well as by the Security Administrator. In the latter case use of TOE functionality for the verification is not mandated, so verification could be done using non-TOE functionality of the device containing the TOE or without using the device containing the TOE.

For distributed TOEs all TOE components shall support Trusted Update. The verification of the signature or hash on the update shall either be done by each TOE component itself (signature verification) or for each component (hash verification).

Updating a distributed TOE might lead to the situation where different TOE components are running different software versions. Depending on the differences between the different software versions the impact of a mixture of different software versions might be no problem at all or critical to the proper functioning of the TOE. The TSS shall detail the mechanisms that support the continuous proper functioning of the TOE during trusted update of distributed TOEs.

Application Note 200

Future versions of this cPP will mandate the use of a digital signature mechanism for trusted updates.

Application Note 201

If certificates are used by the update verification mechanism, certificates are validated in accordance with FIA_X509_EXT.1 and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TUD_EXT.2 must be included in the ST.

Application Note 202

“Update” in the context of this SFR refers to the process of replacing a non-volatile, system resident software component with another. The former is referred to as the NV

image, and the latter is the update image. While the update image is typically newer than the NV image, this is not a requirement. There are legitimate cases where the system owner may want to rollback a component to an older version (e.g. when the component manufacturer releases a faulty update, or when the system relies on an undocumented feature no longer present in the update). Likewise, the owner may want to update with the same version as the NV image to recover from faulty storage.

All discrete firmware and software components (e.g. applications, drivers, and kernel) of the TSF, need to be protected, i.e. they should either be digitally signed by the corresponding manufacturer and subsequently verified by the mechanism performing the update or a hash should be published for them which needs to be verified before the update. Since it is recognized that components may be signed by different manufacturers (in case signatures are used to protect updates), it is essential that the update process verify that both the update and NV images were produced by the same manufacturer (e.g. by comparing public keys) or signed by legitimate signing keys (e.g. successful verification of certificates when using X.509 certificates).

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: No other components.

FPT_TUD_EXT.2.1 The TSF shall not install an update if the code signing certificate is deemed invalid.

FPT_TUD_EXT.2.2 When the certificate is deemed invalid because the certificate has expired, the TSF shall [selection: *allow the Administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate*].

Application Note 203

Certificates may optionally be used for code signing of system software updates (FPT_TUD_EXT.1.3). This element must be included in the ST if certificates are used for validating updates. If “code signing for system software updates” is selected in FIA_X509_EXT.2.1, FPT_TUD_EXT.2 must be included in the ST.

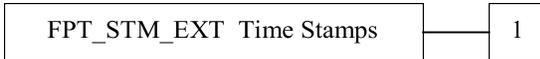
Validity is determined by the certificate path, the expiration date, and the revocation status in accordance with FIA_X509_EXT.1. For expired certificates the author of the ST selects whether the certificate shall be accepted, rejected or the choice is left to the Administrator to accept or reject the certificate.

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*].

Application Note 204

Reliable time stamps are expected to be used with other TSF, e.g. for the generation of audit data to allow the Security Administrator to investigate incidents by checking the order of events and to determine the actual local time when events occurred. The decision about the required level of accuracy of that information is up to the Administrator. The TOE depends on external time and date information, either provided manually by the Security Administrator or through the use of one or more external time sources like NTP servers. The corresponding option(s) shall be chosen from the selection in FPT_STM_EXT.1.2. The use of a local real-time clock and the automatic synchronisation with an external time source (e.g. NTP server) is recommended but not mandated. Note that for the communication with an external time source like an NTP server, the use of FTP_ITC.1 is optional but not mandated. The ST author describes in the TSS how the external time and date information is received by the TOE and how this information is maintained.

The term “reliable time stamps” refers to the strict use of the time and date information, that is provided externally, and the logging of all discontinuous changes to the time settings including information about the old and new time. With this information the real time for all audit data can be determined. Note, that all discontinuous time changes, Administrator actuated or changed via an automated process, must be audited. No audit is needed when time is changed via use of kernel or system facilities – such as daytime (3) – that exhibit no discontinuities in time.

For distributed TOEs it is expected that the Security Administrator ensures synchronization between the time settings of different TOE components. All TOE components shall either be in sync (e.g. through synchronisation between TOE components or through synchronisation of different TOE components with external NTP servers) or the offset should be known to the Administrator for every pair of TOE components. This includes TOE components synchronized to different time zones.

C.5 TOE Access (FTA)

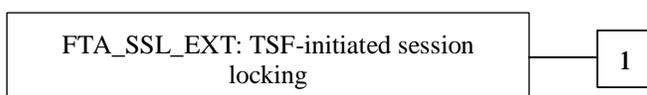
C.5.1 TSF-initiated Session Locking

(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:

- a) 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:

- a) 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.6.1.1 FCO_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.

Application Note 205

This SFR is generally applied to a distributed TOE in order to control the process of creating the distributed TOE from its components by means of a registration process in which a component joins the distributed TOE by registering with an existing component of the distributed TOE. When creating the TSF from the initial pair of components, either of these components may be identified as the TSF for the purposes of satisfying the meaning of “TSF” in this SFR.

The intention of this requirement is to ensure that there is a registration process that includes a positive enablement step by an Administrator before components joining a distributed TOE can communicate with the other components of the TOE and before the new component can act as part of the TSF. The registration process may itself involve communication with the joining component: many network devices use a bespoke process for this, and the security requirements for the “registration communication” are then defined in FCO_CPC_EXT.1.2. Use of this “registration communication” channel is not deemed inconsistent with the requirement of FCO_CPC_EXT.1.1 (i.e. the registration channel can be used before the enablement step, but only in order to complete the registration process).