# Security Target

# FortiAnalyzer™ centralized reporting appliances running Firmware 5.2.4

**Common Criteria Evaluation with**
**Network Device Protection Profile v1.1 Errata #3**

Document Version: 0.8
Date: June 14, 2016

*Prepared For:*

**Fortinet, Inc**

326 Moodie Drive
Ottawa, ON K2H 8G3, Canada
www.fortinet.com

*Prepared By*:

CGI Global IT Security Labs.

1410 Blair Place, 7th floor
Ottawa, ON K1J 9B9, Canada
www.cgi.com/securitylab

# Revision History

| Ver # | Description of changes | Modified by | Date |
|---|---|---|---|
| 0.1 | Initial copy | Danielle Freebourne | July 31, 2014 |
| 0.2 | Updates from lab | Danielle Freebourne | August 27, 2015 |
| 0.3 | ORs and updates from lab | Danielle Freebourne | September 2, 2015 |
| 0.4 | Updates for Errata 3 | Danielle Freebourne | September 25, 2015 |
| 0.5 | ORs and updates from lab | Danielle Freebourne | December 8, 2015 |
| 0.6 | ORs and updates from lab | Danielle Freebourne | December 17, 2015 |
| 0.7 | ORs and updates from lab | Danielle Freebourne | May 9, 2016 |
| 0.8 | ORs and updates from lab | Danielle Freebourne | June 14, 2016 |

# TABLE OF CONTENTS



# LIST OF TABLES



# 1 INTRODUCTION

This section identifies the Security Target (ST), Target of Evaluation (TOE), document conventions, and terminology. It also provides TOE overview and describes the hardware and software that make up the TOE as well as the physical and logical boundaries of the TOE.

## 1.1 ST Reference

| | |
|---|---|
| ST Title | FortiAnalyzer™ centralized reporting appliances running Firmware 5.2.4 |
| ST Revision | 0.8 |
| ST Publication Date | June 14, 2016 |
| ST Author | CGI Global IT Security Labs – Canada |
| | Danielle Freebourne – Junior Consultant |

## 1.2 Target of Evaluation Reference

| | |
|---|---|
| TOE Developer | Fortinet, Inc. |
| TOE Name | FortiAnalyzer™ centralized reporting appliances running Firmware 5.2.4 |
| TOE Version | 5.2.4 Build 0738 FIPS/CC build |

## 1.3 Conventions

The Common Criteria allows for assignment, refinement, selection and iteration operations to be performed on security functional requirements. All of these operations are used within this ST. These operations are performed as described in Part 2 of the CC, and selected presentation choices are discussed below to aid the Security Target reader:

- An assignment operation is indicated by [*italicized text within brackets*].
- Selections are denoted by [underlined text within brackets].
- Refinement of security requirements is identified using **bold text**. Any text removed is indicated with a strikethrough (Example: ~~TSF~~).
- Iterations are identified by appending a number in parentheses following the component title, for example, FIA_UAU.1 (1) and FIA_UAU.1 (2) refer to two iterations of the FIA_UAU.1 security functional requirement component.

## 1.4 TOE Overview

The TOE is the FortiAnalyzer integrated network logging, analysis and reporting appliance running version 5.2 of the Firmware code in stand-alone “FIPS/CC mode”.

The TOE is a log collection and reporting device which relies upon the use of a Web-based manager and a Command Line Interface (CLI) for administrator access. FortiAnalyzer uses Administrative domains (ADOMs) to distinguish amongst administrative TOE units. The TOE works with ADOM and virtual domain (VDOMs) to provide control and constrain accessibility amongst Fortinet devices.

The TOE unit operates in two modes: the Analyzer mode and Collector mode. The Analyzer mode (default mode) has support for all features of FortiAnalyzer, monitoring devices that send logs to the TOE unit for analyzing and reporting. The default mode is used to aggregate logs from a single to many log collectors. The Collector mode provides the capabilities to save and upload logs in their original (binary) formats.

The TOE has extensive logging capabilities, as described by section 6.1.1 of this document. These include, but are not limited to administrative actions and tampering or misuse of the trusted cryptographic channels. The TOE acts as an external audit server over a cryptographically protected channel (TLS) with another IT entity for analysis and inspection of data stored locally.

The TOE implements FIPS and CAVP validated cryptography for all interfaces. Details of the validations are contained within Table 11 and Table 12 of this document. This cryptography is used to secure communications to trusted administrators and to secure audit logs in transit from another IT entity to the TOE. This acts as an external audit server for the purposes of aggregation, analysis and review. User administration sessions are connected over local console and to a remote administrator over HTTPS using validated cryptography. To ensure proper random number generation the TOE has been equipped with a dedicated hardware noise source which provides entropy collected from the ambient environment in which the product operates. This noise source is continually monitored for its ongoing health and proper operation.

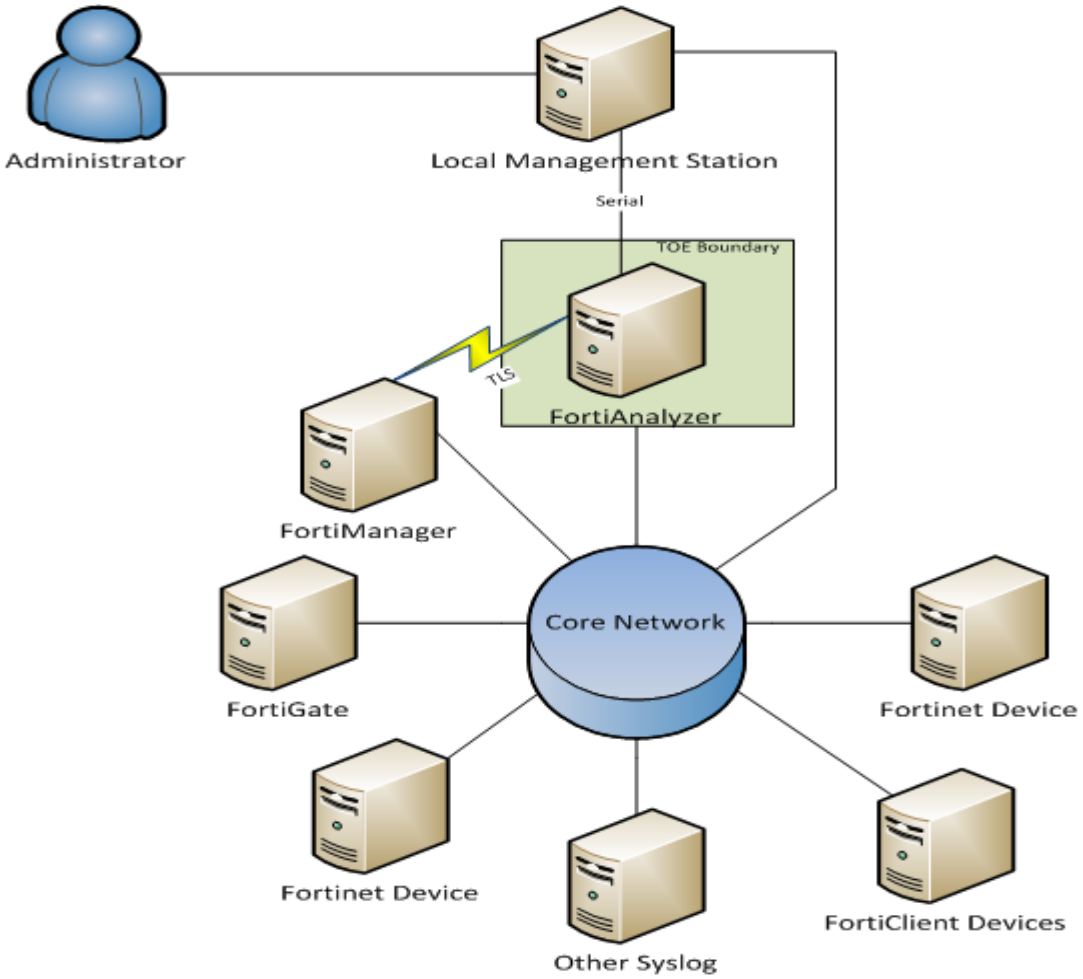
The TOE also offers the ability to verify through cryptographic signatures that product updates are valid, and will reject any updates without the appropriate Fortinet signature. The firmware is inspected on startup as described in the Security Policy of the FIPS 140-2 level 1 in the section detailing the firmware integrity checks.

## 1.5 TOE Description

This section primarily addresses the physical and logical components of the TOE included in the evaluation.

### 1.5.1 Physical Boundary

The physical scope of the TOE includes The TOE hardware as well as the firmware as well as a Fortinet Entropy Token to provide the hardware noise source for conformance with the requirement as defined within the Protection Profile. Details on the TOE boundary, operational environment, and TSFI's are shown below:




#### *1.5.1.1 FortiAnalyzer Hardware Models*

The following TOE hardware platforms are claimed for this evaluation:

- FAZ-200D
- FAZ-1000C
- FAZ-1000D
- FAZ-2000B
- FAZ-3000D
- FAZ-3000E
- FAZ-3500E
- FAZ-3900E
- FAZ-4000B

For the appliances listed above, Section 10 of this document provides a listing of the CPU, memory, and storage capacity used in each model.

#### *1.5.1.2 Guidance Documentation*

The following lists the TOE Guidance Documentation to install, configure, and maintain the TOE.

FortiAnalyzer 5.2.4 Administration Guide 05-524-232167-20150922 September 22, 2015

FortiAnalyzer 5.2.4 CLI Reference 05-523-232152-201150923 September 23, 2015

FortiAnalyzer 5.2.4 Release Notes 05-524-292531-20151102 November 02, 2015

FortiAnalyzer 5.2.4 Upgrade Guide September 22, 2015

FIPS 140-2 and Common Criteria Compliant Operation for FortiAnalyzer 5.2.4

FortiAnalyzer Hardware Manual

### 1.5.2 Logical Boundary

This section outlines the boundaries of the security functionality of the TOE; the logical boundary of the TOE includes the security functionality described in the following sections.

#### *1.5.2.1 Security Audit*

The TOE will generate auditable events as specified in the NDPP which may help indicate a number of potential security concerns including resonance, password guessing and tampering with the trusted paths and channels. For all auditable events the TOE will associate a user (either IP address or with administrative credentials) to the session and use this identifier to be reviewed by administrator to be stored on the TOE.

An authorized administrator may delete the local audit trail. An authorized administrator may configure additional auditable events, configure the back-up of audit data to an external source and manage audit data storage.

The auditing function is supported by reliable timestamps provided by the TOE.

#### *1.5.2.2 Cryptographic Support*

The TOE's cryptographic module is FIPS PUB 140-2 validated and meets Security Level 1 overall. The TOE is capable of generating cryptographic keys using a properly seeded random bit generator in order to provide cryptographic services to the network. The TOE is also capable of importing cryptographic keys and certificates from outside the TOE boundary. These keys are zeroized when no longer required and the TOE offers a function to zeroize these keys on demand.

The TOE is designed such that the cryptographic keys and other critical security parameters are not exposed in plain text through the various interfaces made available to the TOE administrator(s). Passwords including administrative passwords and pre-shared keys are stored on the TOE in the configuration file. These passwords obscure by encrypting the configuration file using an AES-128 key. Certificates are not viewable from any interface and may only be imported to the TOE through HTTPS which is a cryptographically protected trusted and validated channel.

The TOE implements HTTPS and has compatibility with a wide variety of other products.

#### *1.5.2.3 User Data Protection*

The TOE ensures that all information is zeroized on allocation of memory to ensure that all memory is cleared of residual information prior to being written to. Keys and CSP's are zeroized per the FIPS 140-2 module validation.

#### *1.5.2.4 Identification and Authentication*

All administration requires authentication by the user identification and password mechanism. Administration may either be performed locally using the Local Console CLI or remotely using the Network Web-Based GUI. When authenticating locally or remotely the TOE supports complex, configurable password rules and supports complex character sets.

When authenticating over the GUI remote authentication data is protected via an encrypted trusted path between the TOE and administrator. Any individual attempting to log on for an interactive session will be shown a warning message that they must accept prior to being presented with a prompt to attempt their authentication.

#### *1.5.2.5 Security Management*

The TOE provides remote and local administrative interfaces that permit role based administration to configure and manage the TOE both locally and remotely. When fully initialized and configured the TOE is connected to two or more networks and remote administration data flows from a Network Management Station to the TOE. On the TOE hardware model there is also a Local Console which can be connected to from within the physically secured area described within table 7 of the NDPP and consists of a physical serial interface to the TOE.

An administrator account is associated with an access profile, which determines the permissions of the individual administrator. Additionally, each FortiAnalyzer™ install comes with a default administrator account with all permissions, which may not be deleted. The term 'authorized administrator' is used throughout this ST to describe an administrator given the appropriate permission to perform tasks as required.

The TOE has two security management interfaces, including the Local Console CLI with a Network CLU interface and a Web based GUI that uses HTTPS. Through these two interfaces, administrators can configure and manage network systems, log configurations, and reporting configurations.

#### *1.5.2.6 Protection of the TSF*

Inter-TSF communications are protected to ensure availability, confidentiality and detection of modification. This is accomplished through the usage of cryptographic communications for any and all communications with remote IT entities, other components of the TOE and remote administrators. By default detection of modification and audit logging are enabled on TLS connections.

The TOE prevents the reading of all administrator passwords, pre-shared keys, symmetric keys and private keys by obscuring them via encryption. The TOE leverages AES-128 for the encryption of these passwords and keys.

The TOE is capable of querying its current version and displaying it back to the administrator via the trusted interfaces. The TOE also provides a method to verify updates and update the TOE through any of the administrative interfaces. Updates to the TOE software are verified by the TOE during the initial phase of the update process. During this process the TOE verifies that the candidate update is signed by the developer's 2048 bit RSA signature in order to ensure the authenticity of the update. This cryptographic key is used for all FIPS firmware images.

The TOE maintains its own timestamp which is free from outside interference. This timestamp is used for the purposes of generating audit logs and other time-sensitive operations on the TOE including cryptographic key regeneration intervals.

The TOE implements a number of self-test on start-up to ensure the correct operation and configuration of the TOE. These include but are not limited to hardware and entropy source self-tests, checksums of the firmware binaries and correct operation of the FIPS approved cryptographic module. Additionally the TOE maintains ongoing health tests associated with the FIPS cryptographic module and the hardware noise source.

#### *1.5.2.7 TOE Access*

The TOE is capable of terminating both local and remote administrative sessions upon detection of administrator inactivity. The TOE is also capable of terminating a remote session upon request from a remote administrator such as when a request to logout is received.

The TOE provides only administrators with a configurable warning banner prior to initiating any interactive session.

#### *1.5.2.8 Trusted Path/Channels*

A cryptographically protected trusted communications channel is required for all communications with the TOE. For the purposes of transmitting audit data from an authorized IT entity, the TOE is capable of securing the server communications via TLS. The audit data, sent securely from the external IT entity, is then received by the TOE and stored securely on the system. The TOE or the remote peer may initiate this cryptographically protected channel.

The TOE will ensure that HTTPS is used for a trusted path between the TOE and the trusted remote administrator. This path will be used for both the initial administrator authentication and all remote administration requests and is terminated upon session timeout or explicit request from an administrator.

### 1.5.3 Hardware, firmware, and Software Supplied by the IT Environment

The following hardware, firmware and software, which are supplied by the IT environment, are excluded from the TOE boundary.

- FortiManager
- FortiGate
- FortiClient devices
- Other Syslog

- Other Fortinet devices
- Local Serial Console Software
- Supported Web Browser

### 1.5.4 Product Physical/Logical Features and Functions not included in the TOE Evaluation

The FortiAnalyzer™ appliances are capable of a variety of functions and configurations which are not covered by the NDPP. The TOE is capable of this functionality however the following features have not been examined as part of this evaluation:

- Syslog sever logging support
- FortiView module
- Central Quarantine
- SSH (Disabled)
- SNMP
- HTTP (Disabled)
- Web services
- telnet (Disabled)

# 2 CONFORMANCE CLAIMS

## 2.1 Common Criteria Conformance Claim

The Security Target is conformant to Common Criteria Version 3.1 Revision 4, September 2012, Part 2 extended and Part 3 conformant.

## 2.2 Protection Profile Conformance Claim

The Security Target is conformant to the:

- Network Devices Protection Profile (NDPP) v1.1, June 08, 2012, including the following optional requirements [TLS and TLS/HTTPS].
- The NDPP Errata #3, 3 November 2014

# 3 SECURITY PROBLEM DEFINITION

This section defines the security problem which the TOE and its operational environment are supposed to address. Specifically, the security problem makes up the following:

- Any known or assumed threats countered by the TOE or its operational environment.
- Any organizational security policies with which the TOE must comply.
- Any assumptions about the security aspects of the environment and/or of the manner in which the TOE is intended to be used.

This section identifies assumptions as A.*assumption*, threats as T.*threat* and policies as P.*policy*.

## 3.1 Threats

This section identifies the threats to the assets against which protection is required by the TOE or by the security environment. The threat agents are divided into two categories:

- Attackers who are not TOE users: They have public knowledge of how the TOE operates and are assumed to possess a low skill level, limited resources to alter TOE configuration settings or parameters and no physical access to the TOE.
- TOE users: They have extensive knowledge of how the TOE operates and are assumed to possess a high skill level, moderate resources to alter TOE configuration settings or parameters and physical access to the TOE. (TOE users are, however, assumed not to be willfully hostile to the TOE.)

The table below lists threats applicable to the TOE and its operational environment:

Table 1 – Threats

| Threat | Description |
|---|---|
| T.ADMIN_ERROR | An administrator may unintentionally install or configure the TOE incorrectly, resulting in ineffective security mechanisms. |
| T.TSF_FAILURE | Security mechanisms of the TOE may fail, leading to a compromise of the TSF. |
| T.UNDETECTED_ACTIONS | Malicious remote users or external IT entities may take actions that adversely affect the security of the TOE. These actions may remain undetected and thus their effects cannot be effectively mitigated. |
| T.UNAUTHORIZED_ACCESS | A user may gain unauthorized access to the TOE data and TOE executable code. A malicious user, process, or external IT entity may masquerade as an authorized entity in order to gain unauthorized access to data or TOE resources. A malicious user, process or external IT entity may misrepresent itself as the TOE to obtain identification and authentication data. |
| T.UNAUTHORIZED_UPDATE | A malicious party attempts to supply the end user with an update to the product that may compromise the security features of the TOE. |
| T.USER_DATA_REUSE | User data may be inadvertently sent to a destination not intended by the original sender. |

## 3.2 Organizational Security Policies

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. The following table lists Organizational Security Policies (OSP) applicable to the TOE and its operational environment:

Table 2 – Organizational Security Policies

| OSP | 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. |

## 3.3 Assumptions

This section describes the security aspects of the environment in which the TOE is intended to operate. The following specific conditions are assumed to exist in an environment where the TOE is employed.

Table 3 – Assumptions

| Assumption | Description |
|---|---|
| A.NO_GENERAL_PURPOSE | It is assumed that 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. |
| A.PHYSICAL | Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment. |
| A.TRUSTED_ADMIN | TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner. |

# 4 SECURITY OBJECTIVES

Security objectives are concise, abstract statements of the intended solution to the problem defined by the security problem definition. This high-level solution is divided into two parts: the security objectives for the TOE, and the security objectives for the TOE's operational environment. This section identifies the security objectives for the TOE and its supporting environment.

## 4.1 Security Objectives for the TOE

The IT security objectives for the TOE are as follows:

Table 4 – TOE Security Objectives

| Security Objective | Description |
|---|---|
| O.PROTECTED_COMMUNICATIONS | The TOE will provide protected communication channels for administrators, other parts of a distributed TOE, and authorized IT entities. |
| O.VERIFIABLE_UPDATES | The TOE will provide the capability to help ensure that any updates to the TOE can be verified by the Administrator to be unaltered and (optionally) from a trusted source. |
| O.SYSTEM_MONITORING | The TOE will provide the capability to generate audit data and send those data to an external IT entity. |
| O.DISPLAY_BANNER | The TOE will display an advisory warning regarding use of the TOE. |
| O.TOE_ADMINISTRATION | The TOE will provide mechanisms to ensure that only administrators are able to log in and configure the TOE, and provide protections for logged-in administrators. |
| O.RESIDUAL_INFORMATION_CLEARING | The TOE will ensure that any data contained in a protected resource is not available when the resource is reallocated. |
| O.SESSION_LOCK | The TOE shall provide mechanisms that mitigate the risk of unattended sessions being hijacked. |
| O.TSF_SELF_TEST | The TOE will provide the capability to test some subset of its security functionality to ensure it is operating properly. |

## 4.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are addressed below:

Table 5 – Operational Environment Security Objectives

| Security Objective | Description |
|---|---|
| OE.NO_GENERAL_PURPOSE | There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE. |
| OE.TRUSTED_ADMIN | TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner. |

| Security Objective | Description |
|---|---|
| OE.PHYSICAL | Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment. |

# 5 EXTENDED SECURITY REQUIREMENT COMPONENTS DEFINITION

This section defines the extended Security Functional Requirements (SFRs) and extended Security Functional Assurance Requirements (SARs) met by the TOE. All the extended components have been drawn from the Network Device Protection Profile (NDPP) v1.1 and the interpretations and clarifications from NDPP Errata #3.

## 5.1 Extended TOE Security Functional Requirement Components

This section specifies the extended SFRs for the TOE.

### 5.1.1 Class FAU: Security Audit

Families in this class address the requirements for functions to implement security audit as defined in CC Part 2.

#### *5.1.1.1 Family FAU_STG: Security audit event storage*

**Family Behaviour**

This extended family FAU_STG_EXT is modeled after the FAU_STG family. This family defines the requirements for the TSF to be able to create and maintain a secure audit trail. Stored audit records refers to those records within the audit trail, and not the audit records that have been retrieved (to temporary storage) through selection. The requirements of the extended family are focused on the secure transmission of audit records to a remote logging server.

Components in this family address the requirements for protection audit data as defined in CC Part 2. This section defines the extended components for the FAU_STG_EXT family.

**Component Leveling**

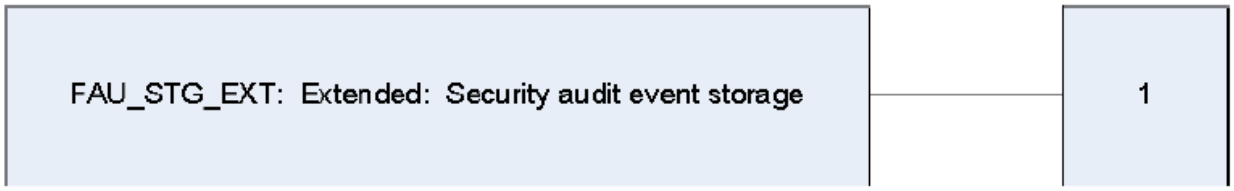



Figure 2 Extended: Security audit event storage family decomposition

FAU_STG_EXT.1 External Audit Trail Storage requires the TSF to use an external IT entity for audit data storage. It is modeled after FAU_STG.1, and is considered to be part of the FAU_STG family.

**Management: FAU_STG_EXT.1**

a) There are no management activities foreseen.

**Audit: FAU_STG_EXT.1**

a) There are no auditable events foreseen.

**FAU_STG_EXT.1 External Audit Trail Storage**

| | |
|---|---|
| Hierarchical to: | No other components |
| Dependencies: | FAU_GEN.1 Audit data generation<br>FTP_ITC.1 Inter-TSF trusted channel |
| **FAU_STG_EXT.1.1** | The TSF shall be able to [selection: transmit the generated audit data to an external IT entity, receive and store audit data from an external IT entity] using a trusted channel implementing the [selection: IPsec, SSH, TLS, TLS/HTTPS] protocol. |

### 5.1.2 Class FCS: Cryptographic Support

Families in this class address the requirements for functions to implement cryptographic functionality as defined in CC Part 2.

#### *5.1.2.1 Family FCS_CKM: Cryptographic Key Management*

**Family Behaviour**

Cryptographic keys must be managed throughout their life cycle. The FCS_CKM family, after which this extended family is modeled, is intended to support that lifecycle and consequently defines requirements for the following activities: cryptographic key generation, cryptographic key distribution, cryptographic key access and cryptographic key destruction. This family should be included whenever there are functional requirements for the management of cryptographic keys. The extended family is designed to include CSP31s and further defines the requirements for plaintext secret and private cryptographic keys. The requirements also further define the key destruction methods allowed, per FIPS 140-2 requirements.

Components in this family address the requirements for managing cryptographic keys as defined in CC Part 2. This section defines the extended components for the FCS_CKM_EXT family.

**Component Leveling**


Figure 3 Extended: Cryptographic key management family decomposition

FCS_CKM_EXT.4 Cryptographic key zeroization requires cryptographic keys and cryptographic critical security parameters to be zeroized. It is modeled after FCS_CKM.4, and is considered to be part of the FCS_CKM family.

**Management: FCS_CKM_EXT.4**

a) There are no management activities foreseen.

**Audit: FCS_CKM_EXT.4**

a) There are no auditable events foreseen.

**FCS_CKM_EXT.4 Cryptographic Key Zeroization**

| | |
|---|---|
| Hierarchical to: | No other components |
| Dependencies: | FDP_ITC.1 Import of user data without security attributes, or<br>FDP_ITC.2 Import of user data with security attributes, or<br>FCS_CKM.1 Cryptographic key generation |
| FCS_CKM_EXT.4.1 | The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required. |

#### *5.1.2.2 Family FCS_HTTPS_EXT: Extended: HTTPS*

**Family Behaviour**

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

**Component Leveling**


Figure 4 Extended: HTTPS family decomposition

FCS_HTTPS_EXT.1 Extended: HTTPS requires that HTTPS be implemented. It belongs to a new family defined for the FCS Class.

**Management: FCS_HTTPS_EXT.1**

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 ST:

a) Failure to establish a HTTPS session, and reason for failure;
b) Establishment/Termination of a HTTPS session, and non-TOE endpoint of connection (IP address) for both successes and failures.

| **FCS_HTTPS_EXT.1** | **Extended: HTTPS** |
|---|---|
| Hierarchical to: | No other components |
| Dependencies: | FCS_TLS_EXT.1 Extended: TLS |
| 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 as specified in FCS_TLS_EXT.1. |

#### *5.1.2.3 Family FCS_TLS_EXT: Extended: TLS*

**Family Behaviour**

Components in this family address the requirements for protecting communications using TLS. This is a new family defined for the FCS Class.

**Component Leveling**

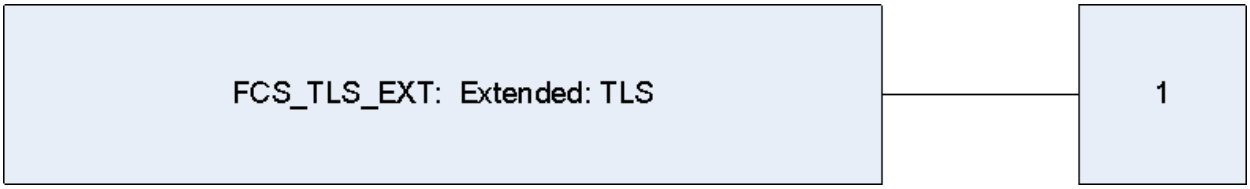



Figure 5 Extended: TLS family decomposition

FCS_TLS_EXT.1 Extended: TLS requires that TLS be implemented. It belongs to a new family defined for the FCS Class.

**Management: FCS_TLS_EXT.1**

a) There are no management activities foreseen.

**Audit: FCS_TLS_EXT.1**

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

a) Failure to establish a TLS session, and reason for failure;
b) Establishment/Termination of a TLS session, and non-TOE endpoint of connection (IP address) for both successes and failures.

| **FCS_TLS_EXT.1** | **Extended: TLS** |
|---|---|
| Hierarchical to: | No other components |
| Dependencies: | FCS_COP.1(1) Cryptographic operation (for data encryption/decryption)<br>FCS_COP.1(2) Cryptographic operation (for cryptographic signature)<br>FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)<br>FCS_COP.1(4) Cryptographic operation (for keyed-hash message authentication)<br>FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)<br>FCS_CKM.1 Cryptographic Key Generation<br>FCS_CKM_EXT.4 Cryptographic Key Zeroization |
| FCS_TLS_EXT.1.1 | The TSF shall implement one or more of the following protocols [selection: TLS 1.0 (RFC 2346), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:<br>**Mandatory Ciphersuites:**<br>TLS_RSA_WITH_AES_128_CBC_SHA<br>TLS_RSA_WITH_AES_256_CBC_SHA<br>TLS_DHE_RSA_WITH_AES_128_CBC_SHA<br>TLS_DHE_RSA_WITH_AES_256_CBC_SHA<br>**Optional Ciphersuites:**<br>[selection:<br>None<br>TLS_RSA_WITH_AES_128_CBC_SHA256<br>TLS_RSA_WITH_AES_256_CBC_SHA256<br>TLS_DHE_RSA_WITH_AES_128_CBC_SHA256<br>TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 |

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
].

#### *5.1.2.4 Family FCS_RBG_EXT: Extended: Random Bit Generation*

**Family Behaviour**

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

**Component Leveling**


Figure 6 Extended: Random Bit Generation family decomposition

FCS_RBG_EXT.1 Extended: Random Bit Generation is the only component of this class. This component requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source. It was modeled after FCS_COP.1 Cryptographic operation.

**Management: FCS_RBG_EXT.1**

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) There are no auditable events foreseen.

| **FCS_RBG_EXT.1** | **Extended: Random Bit Generation (Random bit generation)** |
|---|---|
| Hierarchical to: | No other components |
| Dependencies: | No dependencies |
| FCS_RBG_EXT.1.1 | The TSF shall perform all random bit generation (RBG) services in accordance with [selection, choose one of: NIST[1] Special Publication 800-90 using [selection: Hash_DRBG[2] (any), HMAC[3]_DRBG (any), CTR[4]_DRBG (AES20), Dual_EC[5]_DRBG (any)]; FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES] seeded by an entropy source that accumulated entropy from [selection, one or both of a software-based noise source; a TSF-hardware-based noise source]. |

[1] NIST – National Institute of Standards and Technology
[2] DRBG – Deterministic Random Bit Generator
[3] HMAC – Hashed Message Authentication Code
[4] CTR – Counter Mode
[5] EC – Elliptical Curve

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [selection, choose one of: 128 bits, 256 bits] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

### 5.1.3 Class FIA: Identification and Authentication

Families in this class address the requirements for functions to establish and verify a claimed user identity as defined in CC Part 2.

#### *5.1.3.1 Family FIA_PMG_EXT: Password Management*

**Family Behaviour**

This family defines the password strength rules enforced by the TSF.

This section defines the extended components for the FIA_PMG_EXT family, which is modeled after FIA_SOS Specification of secrets.

**Component Leveling**


Figure 7 Extended: Password Management family decomposition

FIA_PMG_EXT.1 Password Management defines the password strength requirements that the TSF will enforce. It belongs to a new family defined for FIA class.

**Management: FIA_PMG_EXT.1**

a) There are no management activities foreseen.

**Audit: FIA_PMG_EXT.1**

a) There are no auditable events foreseen.

| **FIA_PMG_EXT.1** | **Extended: Password Management** |
|---|---|
| Hierarchical to: | No other components |
| Dependencies: | None |
| FIA_PMG_EXT.1.1 | The TSF shall provide the following password management capabilities for administrative passwords:<br>1. Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and special characters: [selection: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [assignment: *other characters*]];<br>2. Minimum password length shall settable by the Security Administrator, and support passwords of 15 characters or greater; |

#### *5.1.3.2 Family FIA_UAU_EXT: Extended: User Authentication*

**Family Behaviour**

This family defines the types of user authentication mechanisms supported by the TSF.

This section defines the extended components for the FIA_UAU_EXT family, which is modeled after the FIA_UAU User authentication family.

**Component Leveling**


Figure 8 Extended: User authentication family decomposition

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism requires a local password-based authentication mechanism and the capability for passwords to expire. In addition, other authentication mechanisms can be specified. It is considered to be part of the FIA_UAU family.

**Management: FIA_UAU_EXT.2**

a) There are no management activities foreseen.

**Audit: FIA_UAU_EXT.2**

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

a) All use of the authentication mechanisms.

| **FIA_UAU_EXT.2** | **Extended: Password-based Authentication Mechanism** |
|---|---|
| Hierarchical to: | No other components |
| Dependencies: | None |
| FIA_UAU_EXT.2.1 | The TSF shall provide a local password-based authentication mechanism, [selection: [assignment: *other authentication mechanism(s)*], none] to perform user authentication. |

### *5.1.3.3 Family FIA_UIA_EXT: Extended: User Identification and Authentication*

**Family Behaviour**

This family defines the types of user identification and authentication mechanisms supported by the TSF.

This section defines the components for the extended FIA_UIA_EXT family, which is modeled after the FIA_UAU and FIA_UID families.

**Component Leveling**

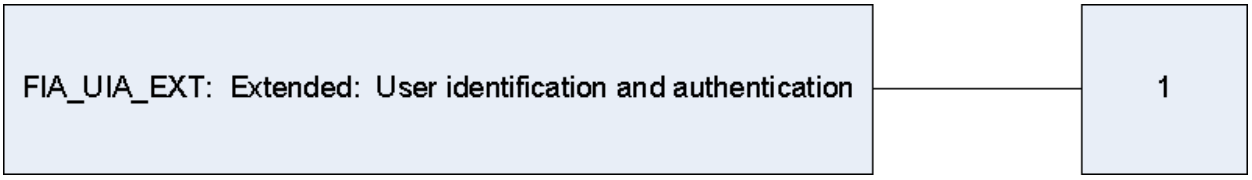



Figure 9 Extended: User identification and authentication family decomposition

FIA_UIA_EXT.1 Extended: Password-based Authentication and Identification Mechanism, requires a local password-based authentication mechanism and the capability for passwords to expire. In addition, other authentication mechanisms can be specified. It is based on a combination of FIA_UAU.1 and FIA_UID.1, and belongs to a new family defined for class FIA.

**Management: FIA_UIA_EXT.1**

a) There are no management activities foreseen.

**Audit: FIA_UIA_EXT.1**

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

a) All use of the authentication mechanism with provided user identity and origin of the attempt (e.g. IP address).

| **FIA_UIA_EXT.1** | **Extended: Password-based Authentication and Identification Mechanism** |
|---|---|
| Hierarchical to: | FIA_UID.1 Timing of identification<br>FIA_UAU.1 Timing of Authentication |
| Dependencies: | None |
| 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:<br>○ Display the warning banner in accordance with FTA_TAB.1;<br>○ [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. |

### 5.1.4 Class FPT: Protection of the TSF

Families in this class address the requirements for functions providing integrity and management of mechanisms that constitute the TSF and of the TSF data as defined in CC Part 2.

#### *5.1.4.1 Family FPT_APW_EXT: Extended: Protection of Administrator Passwords*

**Family Behaviour**

Components in this family address the requirements for managing and protecting TSF data, such as passwords. This is a new family modeled after the FPT_PTD family.

**Component Leveling**

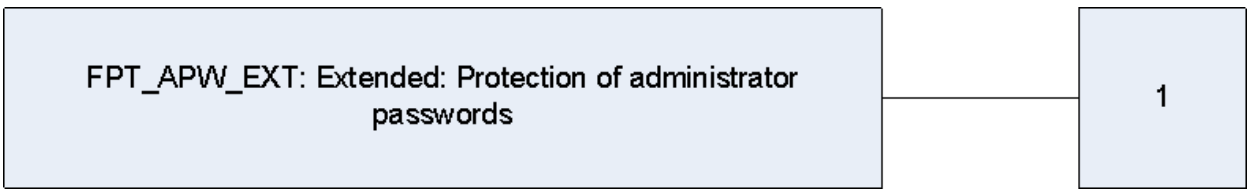



Figure 10 Extended: Protection of administrator passwords family decomposition

FPT_APW_EXT.1 Extended: Protection of Administrator Passwords requires administrator passwords to be stored in non-plaintext form and requires the TOE to prevent reading of plaintext passwords. It is modeled after FPT_SSP.2, but it belongs to a new family defined for the FPT class.

**Management: FPT_APW_EXT.1**

a) There are no management activities foreseen.

**Audit: FPT_APW_EXT.1**

a) There are no audit activities foreseen.

| **FPT_APW_EXT.1** | **Extended: Protection of Administrator Passwords** |
|---|---|
| Hierarchical to: | No other components |
| Dependencies: | None |
| 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. |

### *5.1.4.2 Family FPT_SKP_EXT: Extended: Protection of TSF Data*

**Family Behaviour**

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

**Component Leveling**

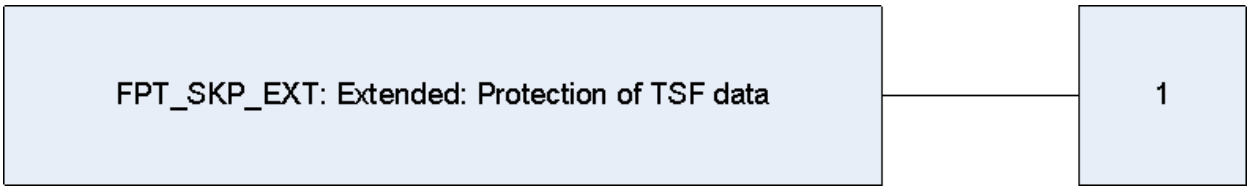



Figure 11 Extended: Protection of TSF data family decomposition

FPT_SKP_EXT.1 Extended: Protection of TSF data (for reading of all symmetric keys) requires the TOE to prevent reading of all pre-shared, symmetric, and private keys. It is modeled after FPT_SSP.1, but it belongs to a new family defined for the FPT class.

**Management: FPT_SKP_EXT.1**

a) There are no management activities foreseen.

**Audit: FPT_SKP_EXT.1**

a) There are no audit activities foreseen.

| **FPT_SKP_EXT.1** | **Extended: Protection of TSF data (for reading of all symmetric keys)** |
|---|---|
| Hierarchical to: | No other components |
| Dependencies: | None |
| FPT_SKP_EXT.1.1 | The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys. |

### *5.1.4.3 Family FPT_TST_EXT: Extended: TSF Self Test*

**Family Behaviour**

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

The extended FPT_TST_EXT family is modeled after the FPT_TST family.

**Component Leveling**

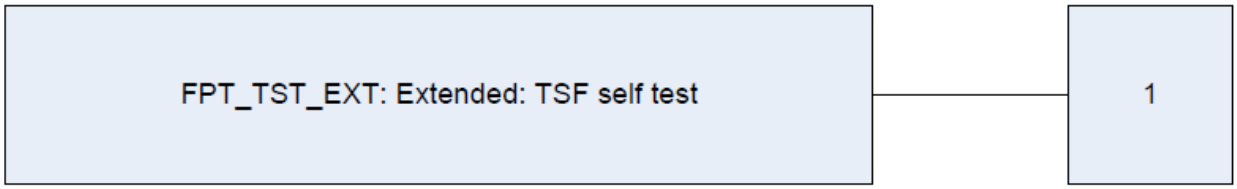



Figure 12 Extended: TSF testing family decomposition

FPT_TST_EXT.1 Extended: TSF testing requires a suite of self-tests to be run during initial start-up in order to demonstrate correct operation of the TSF. It is modeled after FPT_TST.1, but belongs to a new family defined for class FPT.

**Management: FPT_TST_EXT.1**

a) There are no management activities foreseen.

**Audit: FPT_TST_EXT.1**

a) There are no audit activities foreseen.

| **FPT_TST_EXT.1** | **Extended: TSF testing** |
|---|---|
| Hierarchical to: | No other components |
| Dependencies: | None |
| FPT_TST_EXT.1.1 | The TSF shall run a suite of self-tests during initial start-up (on power on) to demonstrate the correct operation of the TSF. |

### *5.1.4.4 Family FPT_TUD_EXT: Extended: Trusted Update*

**Family Behaviour**

Components in this family address the requirements for updating the TOE firmware and/or software. This is a new family defined for the FPT Class.

**Component Leveling**


Figure 13 Extended: Trusted update family decomposition

FPT_TUD_EXT.1 Extended: Management of TSF Data, requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation. It belongs to a new family defined for the FPT class.

**Management: FPT_ TUD_EXT.1**

a) There are no management activities foreseen.

**Audit: FPT_ TUD_EXT.1**

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

a) Initiation of update.

**FPT_ TUD_EXT.1 Extended: Trusted Update**

| | |
|---|---|
| Hierarchical to: | No other components |
| Dependencies: | FCS_COP.1(2) Cryptographic operation (for cryptographic signature)<br>FCS_COP.1(3) Cryptographic operation (for cryptographic hashing) |
| FPT_TUD_EXT.1.1 | The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software. |
| FPT_TUD_EXT.1.2 | The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software. |
| FPT_TUD_EXT.1.3 | The TSF shall provide a means to verify firmware/software updates to the TOE using a [selection: digital signature mechanism, published hash] prior to installing those updates. |

### 5.1.5 Class FTA: TOE Access

Families in this class specify functional requirements for controlling the establishment of a user's session as defined in CC Part 2.

#### *5.1.5.1 Family FTA_SSL_EXT: Extended: TSF-initiated Session Locking*

**Family Behaviour**

Components in this family address the requirements for managing and protecting TSF data, such as passwords. This is a new family modeled after the FPT_PTD family.

**Component Leveling**


Figure 14 Extended: TSF-initiated session locking family decomposition

FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking requires system initiated locking of an interactive session after a specified period of inactivity. It is part of the FTA_SSL 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 ST:

a) Any attempts at unlocking an interactive session.

**FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking**

| | |
|---|---|
| Hierarchical to: | No other components |
| Dependencies: | FIA_UIA_EXT.1 Password-based Authentication and Identification Mechanism |
| FTA_SSL_EXT.1.1 | The TSF shall, for local interactive sessions, [selection:<br>• lock the session – disable any activity of the user'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;<br>• terminate the session]<br>after a Security Administrator-specified time period of inactivity. |

## 5.2 Extended TOE Security Assurance Requirement Components

There are no extended TOE Security Assurance Requirement Components.

# 6 SECURITY REQUIREMENTS

This section defines the Security Functional Requirements (SFRs) and Security Functional Assurance Requirements (SARs) met by the TOE. All the components have been drawn from the Network Device Protection Profile (NDPP) v1.1, Errata #3 of the NDPP and clarifications and interpretations made with NDPP Errata #3.

## 6.1 Security Functional Requirements

The functional security requirements for this Security Target consist of the components from Part 2 of the CC, and those that were explicitly stated, all of which are summarized in the following table:

Table 6 – TOE Security Functional Requirements

| Requirement Class | Requirement Name | Description |
|---|---|---|
| FAU<br>Security Audit | FAU_GEN.1 | Audit Data Generation |
| | FAU_GEN.2 | User Identity Association |
| | FAU_STG_EXT.1 | External Audit Trail Storage |
| FCS<br>Cryptographic support | FCS_CKM.1 | Cryptographic key generation (for asymmetric keys) |
| | FCS_CKM_EXT.4 | Cryptographic Key Zeroization |
| | FCS_COP.1(1) | Cryptographic operation (for data encryption/decryption) |
| | FCS_COP.1(2) | Cryptographic operation (for cryptographic signature) |
| | FCS_COP.1(3) | Cryptographic operation (for cryptographic hashing) |
| | FCS_COP.1(4) | Cryptographic operation (for keyed-hash message authentication) |
| | FCS_HTTPS_EXT.1 | Explicit: HTTPS |
| | FCS_RBG_EXT.1 | Extended: Cryptographic Operation (Random Bit Generation) |
| | FCS_TLS_EXT.1 | Explicit: TLS |
| FDP<br>User Data Protection | FDP_RIP.2 | Full Residual Information Protection |
| FAI<br>Identification and Authentication | FIA_PMG_EXT.1 | Password Management |
| | FIA_UAU.7 | Protected Authentication Feedback |
| | FIA_UAU_EXT.2 | Extended: Password-based Authentication Mechanism |
| | FIA_UIA_EXT.1 | User Identification and Authentication |
| FMT<br>Security Management | FMT_MTD.1 | Management of TSF data (for general TSF data) |
| | FMT_SMF.1 | Specification of management functions |
| | FMT_SMR.2 | Restrictions on Security Roles |
| FPT<br>Protection of the TSF | FPT_APW_EXT.1 | Extended: Protection of Administrator Passwords |
| | FPT_SKP_EXT.1 | Extended: Protection of TSF Data (for reading of all symmetric keys) |
| | FPT_STM.1 | Reliable Time Stamps |

| Requirement Class | Requirement Name | Description |
|---|---|---|
| | FPT_TST_EXT.1 | TSF testing |
| | FPT_TUD_EXT.1 | Extended: Trusted Update |
| FTA<br>TOE Access | FTA_SSL.3 | TSF-initiated Termination |
| | FTA_SSL.4 | User-initiated Termination |
| | FTA_SSL_EXT.1 | TSF-initiated session locking |
| | FTA_TAB.1 | Default TOE access banners |
| FTP<br>Trusted Path/Channels | FTP_ITC.1 | Inter-TSF Trust Channel |
| | FTP_TRP.1 | Trusted Path |

### 6.1.1 Security Audit (FAU)

#### *6.1.1.1 FAU_GEN.1 Audit Data Generation*

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the [not specified] level of audit;
c) [*All administrative actions*]; and
d) [*Specifically defined auditable events listed in Table 7*]

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

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

Table 7 – Auditable Events

| Requirements | 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_EXT.4 | None | None |
| FCS_COP.1(1) | None | None |
| FCS_COP.1(2) | None | None |
| FCS_COP.1(3) | None | None |
| FCS_COP.1(4) | None | None |
| FCS_RBG_EXT.1 | None | None |
| FCS_TLS_EXT.1 | Failure to establish a TLS Session<br>Establishment/Termination of a TLS session | Reason for failure |

| Requirements | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| | | Non-TOE endpoint of connection (IP address) for both successes and failures |
| FCS_HTTPS_EXT.1 | Failure to establish a HTTPS Session Establishment/Termination of a HTTPS session | Reason for failure<br>Non-TOE endpoint of connection (IP address) for both successes and failures |
| FDP_RIP.2 | None | None |
| FIA_PMG_EXT.1 | None | None |
| FIA_UIA_EXT.1 | All use of the identification and authentication mechanism | Provided user identity, origin of the attempt (e.g., IP address) |
| FIA_UAU_EXT.2 | All use of the authentication mechanism | Origin of the attempt (e.g., IP address ) |
| FIA_UAU.7 | None | None |
| FMT_MTD.1 | None | None |
| FMT_SMF.1 | None | None |
| FMT_SMR.2 | None | None |
| FPT_SKP_EXT.1 | None | None |
| FPT_APW_EXT.1 | None | None |
| FPT_STM.1 | Changes to the time | The old and new values for the time Origin of the attempt (e.g. IP address) |
| FPT_TUD_EXT.1 | Initiation of update | No additional information |
| FPT_TST_EXT.1 | None | None |
| FTA_SSL_EXT.1 | Any attempts at unlocking of an interactive session | No additional information |
| FTA_SSL.3 | The termination of a remote session by the session locking mechanism | No additional information |
| FTA_SSL.4 | The termination of an interactive session | No additional information |
| FTA_TAB.1 | None | None |
| FTP_TRP.1 | Initiation of the trusted path. Termination of the trusted path. Failures of the trusted path functions | Identification of the claimed user identity |
| FTP_ITC.1 | Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions | Identification of the initiator and target of failed trusted channels establishment attempt |

#### *6.1.1.2 FAU_GEN.2 User Identity Association*

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

#### *6.1.1.3 FAU_STG_EXT.1 External Audit Trail Storage*

FAU_STG_EXT.1.1 The TSF shall be able to [receive and store audit data from an external IT entity] using a trusted channel implementing the [TLS] protocol.

## 6.1.2 Cryptographic Support (FCS)

### *6.1.2.1 FCS_CKM.1 Cryptographic Key Generation*

FCS_CKM.1.1 The TSF shall generate **asymmetric** cryptographic keys **used for key establishment** in accordance with

- *NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography" for RSA-based key establishment schemes*]

and specified cryptographic key sizes [*equivalent to, or greater than, a symmetric key strength of 112 bits*].

### *6.1.2.2 FCS_CKM_EXT.4 Cryptographic Key Zeroization*

FCS_CKM_EXT.4.1 The TSF shall zeroize all plaintext secret and private cryptographic keys and CSPs when no longer required.

### *6.1.2.3 FCS_COP.1(1) Cryptographic Operation (for data encryption/decryption)*

FCS_COP.1.1(1) Refinement: The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [*AES operating in [CBC[6]]*] and cryptographic key sizes *128-bits and 256-bits that meets the following:*

- ***FIPS PUB 197, "Advanced Encryption Standard (AES)"***
- ***[NIST SP 800-38A]***

### *6.1.2.4 FCS_COP.1(2) Cryptographic operation (for cryptographic signature)*

FCS_COP.1.1(2) Refinement: The TSF shall perform **cryptographic signature services** in accordance with a
***(2) RSA Digital Signature Algorithm (rDSA) with a key size (modulus) of 2048 bits or greater, or***

that meets the following:

**Case: RSA Digital Signature Algorithm**

- **FIPS PUB 186-2 or FIPS PUB 186-3, "Digital Signature Standard"**

### *6.1.2.5 FCS_COP.1(3) Cryptographic Operation (for cryptographic hashing)*

FCS_COP.1.1(3) Refinement: The TSF shall perform [*cryptographic hashing services*] in accordance with a specified cryptographic algorithm **[SHA-1, SHA-256] and message digest sizes [160, 256] bits** that meet the following: *FIPS Pub 180-3, "Secure Hash Standard."*

[6] CBC: Cipher Block Chaining

#### *6.1.2.6 FCS_COP.1(4) Cryptographic Operation (for keyed-hash message authentication)*

FCS_COP.1.1(4) Refinement: The TSF shall perform [*keyed-hash message authentication*] in accordance with a specified cryptographic algorithm HMAC-**[SHA-1, SHA-256], key size [*160, 256*], and message digest sizes [160, 256] bits** that meet the following: *FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code, and FIPS Pub 180-3, "Secure Hash Standard."*

#### *6.1.2.7 FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)*

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [NIST Special Publication 800-90 using [CTR_DRBG (AES)]] seeded by an entropy source that accumulated entropy from [a TSF-hardware-based noise source].

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded with a minimum of [256 bits] of entropy at least equal to the greatest security strength of the keys and hashes that it will generate.

#### *6.1.2.8 FCS_HTTPS_EXT.1 Extended: HTTPS*

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 as specified in FCS_TLS_EXT.1.

#### *6.1.2.9 FCS_TLS_EXT.1 Extended: TLS*

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols [TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246] supporting the following ciphersuites:

**Mandatory Ciphersuites:**

TLS_RSA_WITH_AES_128_CBC_SHA

**Optional Ciphersuites:**

[TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA].

### 6.1.3 User Data Protection (FDP)

#### *6.1.3.1 FDP_RIP.2 Full Residual Information Protection*

FDP_RIP.2.1 The TSF shall ensure that any previous information content of a resource is made unavailable upon the [allocation of the resource to] all objects.

### 6.1.4 Identification and Authentication (FIA)

#### *6.1.4.1 FIA_PMG_EXT.1 Password Management*

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

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

#### *6.1.4.2 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.

#### *6.1.4.3 FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism*

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [none] to perform administrative user authentication.

#### *6.1.4.4 FIA_UAU.7 Protected Authentication Feedback*

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

### 6.1.5 Security Management (FMT)

#### *6.1.5.1 FMT_MTD.1 Management of TSF Data (for general TSF data)*

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

#### *6.1.5.2 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 update the TOE, and to verify the updates using [selection: digital signature] capability prior to installing those updates*[7]*;*
- *[Ability to configure the cryptographic functionality;]*

[7] FortiOS requires that the candidate firmware upgrade be uploaded to the TOE prior to the digital signature being validated

#### *6.1.5.3 FMT_SMR.2 Restrictions on Security Roles*

FMT_SMR.2.1 The TSF shall maintain the roles: [*Authorized 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 [

- *Authorized Administrator role shall be able to administer the TOE locally;*
- *Authorized Administrator role shall be able to administer the TOE remotely*;]

are satisfied.

### 6.1.6 Protection of the TSF (FPT)

#### *6.1.6.1 FPT_SKP_EXT.1 Extended: Protection of TSF Data (for reading of all symmetric keys)*

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

#### *6.1.6.2 FPT_APW_EXT.1 Extended: 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.

#### *6.1.6.3 FPT_STM.1 Reliable Time Stamps*

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

#### *6.1.6.4 FPT_TUD_EXT.1 Extended: Trusted Update*

FPT_TUD_EXT.1.1 The TSF shall provide security administrators the ability to query the current version of the TOE firmware/software.

FPT_TUD_EXT.1.2 The TSF shall provide security administrators the ability to initiate updates to TOE firmware/software.

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

#### *6.1.6.5 FPT_TST_EXT.1: TSF Testing*

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

### 6.1.7 TOE Access (FTA)

#### *6.1.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking*

FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [terminate the session] after a Security Administrator-specified time period of inactivity.

#### *6.1.7.2 TSF-initiated Termination*

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

#### *6.1.7.3 FTA_SSL.4 User-initiated Termination*

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

#### *6.1.7.4 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.

### 6.1.8 Trusted Path/Channels (FTP)

#### *6.1.8.1 FTP_ITC.1 Inter-TSF trusted channel*

FTP_ITC.1.1 The TSF shall **use [TLS]** to provide a **trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: audit server, [transmitting data]** 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 [to receive logs, IPS Packet Logs and reports].

#### *6.1.8.2 FTP_TRP.1 Trusted Path*

FTP_TRP.1.1 The TSF shall **use [TLS/HTTPS]** provide a **trusted** communication path between itself and **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 **detection of modification of the communicated data**].

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

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

## 6.2 Security Assurance Requirements

This section defines the Security Assurance Requirements (SARs) for the TOE. The assurance requirements are taken from NDPP v1.1. The assurance components are summarized in the following table:

Table 8 – Security Assurance Requirements

| Assurance Classes | Assurance Component | Description |
|---|---|---|
| Security Target | ASE_CCL.1 | Conformance claims |
| | ASE_ECD.1 | Extended components definition |
| | ASE_INT.1 | ST introduction |
| | ASE_OBJ.1 | Security objectives |
| | ASE_REQ.1 | Security requirements |
| | ASE_TSS.1 | TOE summary specification |
| Development | ADV_FSP.1 | Basic Functional Specification |
| Guidance Documents | AGD_OPE.1 | Operational User Guidance |
| | AGD_PRE.1 | Preparative Procedures |
| Lifecycle Support | ALC_CMC.1 | Labeling of the TOE |
| | ALC_CMS.1 | TOE CM Coverage |
| Tests | ATE_IND.1 | Independent Testing – Conformance |
| Vulnerability Assessment | AVA_VAN.1 | Vulnerability Analysis |

# 7 TOE SUMMARY SPECIFICATION

This section presents information to detail how the TOE meets the security functional requirements described in previous sections of this ST.

## 7.1 Security Audit

For all administrative actions and events, the TOE creates audit records. The TOE records time of the event, the identity of the administrator or user who caused the event and details of the event as they occur. A user with the appropriate privileges will be able to view audited records while only administrators with appropriate privileges can delete audit records. Audit records cannot be modified. When an action identified in Table 9 is triggered, the IT entity will transcribe the event including the date and time, administrative username of the user triggering the event and the success or failure of the event to the TOE then stored locally.

Table 9 – Auditable Events

| Requirements | Auditable Events | Additional Audit Record Contents |
|---|---|---|
| FCS_TLS_EXT.1 | Failure to establish a TLS Session<br>Establishment/Termination of a TLS session | Reason for failure<br>Non-TOE endpoint of connection (IP address) for both successes and failures |
| FCS_HTTPS_EXT.1 | Failure to establish a HTTPS Session<br>Establishment/Termination of a HTTPS session | Reason for failure<br>Non-TOE endpoint of connection (IP address) for both successes and failures |
| FIA_UIA_EXT.1 | All use of the identification and authentication mechanism | Provided user identity, origin of the attempt (e.g., IP address) |
| FIA_UAU_EXT.2 | All use of the authentication mechanism | Origin of the attempt (e.g., IP address ) |
| FPT_STM.1 | Changes to the time | The old and new values for the time Origin of the attempt (e.g. IP address) |
| FPT_TUD_EXT.1 | Initiation of update | No additional information |
| FTA_SSL_EXT.1 | Any attempts at unlocking of an interactive session | No additional information |
| FTA_SSL.3 | The termination of a remote session by the session locking mechanism | No additional information |
| FTA_SSL.4 | The termination of an interactive session | No additional information |
| FTP_TRP.1 | Initiation of the trusted path. Termination of the trusted path. Failures of the trusted path functions | Identification of the claimed user identity |
| FTP_ITC.1 | Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted channel functions | Identification of the initiator and target of failed trusted channels establishment attempt |

The TOE provides timestamps using an internal clock that is set by an administrator, with changes logged.

The TOE is capable of generating events for external entities based on logging filters. These entities may include an email destination, an SNMP server, or a syslog server.

The TOE is capable of receiving and creating logs which are listed in Table 9. These events include attempts to establish or terminate sessions and errors detected during decryption or validation of data.

Audit records are stored locally using memory, a hard disk, or a FLASH memory card depending on the model. The full list of storage capacity for each device can be found in Table 10. To prevent the loss of audits due to the log capacity being reached, an administrator can configure the TOE to overwrite the oldest audit records once capacity has been reached.

Table 10 – Device Storage

| Model | Flash | Storage |
|---|---|---|
| FAZ-200D | 2 GB | 1 TB |
| FAZ-1000C | 2 GB | 2TB x 1 (Max 8 TB) |
| FAZ-1000D | 2GB | 8 TB |
| FAZ-2000B | None | 2 TB (Max 6 TB) |
| FAZ-3000D | 2GB | 8 X 2TB |
| FAZ-3000E | 2GB | 16 TB |
| FAZ-3500E | 2GB | 24 TB (Max 48 TB) |
| FAZ-3900E | 2 GB | 15 TB |
| FAZ-4000B | None | 6 TB (P24 TB Optional, 16 TB File System) |

The TSF is capable of monitor and logging messages to the audit log for interactions which occur on the remote interfaces. These events include attempts to establish or terminate sessions and errors detected during decryption or validation of data.

If assistance is needed on the interpretation of the audit records, the administrator can request the vendor provide a document[8] that provides clarification on each event. The Log Reference document provides information including Log ID, severity, subcategory, and a high-level description along with an explanation of each name, variable type, and description of the contents of the audit.

## 7.2 Cryptographic Support

The TOE uses FIPS-approved cryptography that has been implemented in the FIPS 140-2 validated cryptographic module. The FIPS-validated cryptographic module implemented in the TSF meets Security Level 1 overall. Cryptographic key destruction by the TOE meets the key zeroization requirements of Key Management Security Level 1 from FIPS PUB 140-2. The TOE only stores ephemeral keys in memory, either in RAM or Flash memory. Persistent keys are encrypted using AES-128 with a key generated when the TOE is initialized and written to the TOE configuration file. The AES key cannot be viewed or backed up through any of the TOE interfaces. The configuration file can be exported or backed up with the passwords remaining in the encrypted format.

[8] FortiManager 5.2.4 Build 738 Log Definition

Cryptographic keys for the TOE which are no longer required are destroyed by overwriting the key storage area with an alternating pattern at least once. Session requests that use these keys are received, the encryption function is run on them and the obscured results are compared with the value in the configuration file. The credential provided is then zeroized through overwriting the storage area following the completion of the authentication request.

The TOE is capable of generating 256 bits of entropy using a dedicated hardware noise source and using this to seed random bit generator in order to provide cryptographic services with up to 256 bits of strength. The TOE is also capable of importing cryptographic keys and certificates from outside the TOE boundary. These keys are zeroized when no longer required and the TOE offers a function to zeroize these keys on demand. For additional information please reference CMVP validations # 2526.

A detailed design of the cryptographic subsystems and entropy noise sources provided by the TOE has been conducted and was used to design the TOE to ensure strong seeding of the DRBG.

The following certificates have been issued by the CMVP and CAVP for Firmware 5.2 and are implemented accordingly in the TOE.

Table 11 – FortiAnalyzer[9] Cryptographic Module Algorithms

| Cryptographic Operations | Cryptographic Algorithm | Key Sizes (bits) | Message Digest Size | FIPS Standard | Certificate # |
|---|---|---|---|---|---|
| Symmetric Encryption and Decryption | AES operating in CBC | 128, 256 | N/A | FIPS PUB 197 (AES)<br>NIST SP800-38A | CAVP Certificate # (3595) |
| Random Bit Generation | CTR DRBG | 256 | N/A | NIST SP 800-90A | CAVP Certificate # (930) |
| Cryptographic Hashing | SHA-1, SHA-256 | 160, 256 | 160, 256 | FIPS Pub 198-1 (HMAC)<br>FIPS Pub 180-3 (SHS) | CAVP Certificate # (2957) |
| Keyed-Hash message authentication (HMAC) | SHA-1, SHA-256 | 160, 256 | 160, 256 | FIPS Pub 180-3 (SHS) | CAVP Certificate # (2292) |
| Signature Verification | rDSA | 2048, 3072 | N/A | FIPS PUB 186-3 (DSS)<br>FIPS PUB 186-2 (DSS) | CAVP Certificate # (1849) |
| Signature Generation | rDSA | 2048, 3072 | N/A | FIPS PUB 186-3 (DSS)<br>FIPS PUB 186-2 (DSS) | CAVP Certificate # (1849) |

[9] While the name of this module is “SSL” it is used for HTTPS and may be used for some ciphers in TLS connections. This module has undergone CAVP validation and corresponds with the name provided on CAVP certificate.

As part of the CMVP testing the FIPS lab validated the implementation for both the “DRBG” and “SSL” cryptographic libraries via the CAVS tool for the correctness of the implementation. These libraries are included within the CMVP certificate for the FIPS 140-2 Level 1 validation. For additional details on the cryptographic operations, it is encouraged to reference the appropriate certificate as stated inTable 11.

### 7.2.1 Entropy Source and Random Bit Generation

The TOE implements an entropy collection system from a hardware based noise source which is derived from wide-band RF white noise which is then pooled and conditioned prior to being used. The raw unconditioned noise from this source has been analyzed using a Fortinet supplied development build based on the TOE version. This data was analyzed using a NIAP supplied entropy test tool to determine the entropy rate present in the samples. This analysis confirmed the claim that during typical operating conditions the TOE generates appropriate entropy for the random number generation to provide protection up to 256 bits of strength.

The Fortinet Cryptographic Module contains a CTR_DRBG implemented per NIST SP 800-90A and is seeded with an entropy source derived from the hardware source. Entropy from the noise source are extracted and conditioned through a NIST SP 800-90B approved conditioning function to seed the DRBG 256 bits of entropy. A failure of the entropy source is a blocking event for the reseeding of the cryptographic system and the health of the entropy source is continually monitored. The noise source health tests are constructed to ensure that a catastrophic failure of the noise source will keep the TOE from reseeding with weak entropy and a prolonged failure will halt the operation of the TOE. The CTR_DRBG implementation has been CAVP tested to ensure correct operation.

### 7.2.2 Cryptographically Trusted Paths

Trusted paths are used to protect remote administrator authentication and all remote administrator actions. Remote administration sessions apply to the Network Web-Based GUI (HTTPS) only.

### 7.2.3 HTTPS

The Network Web-Based GUI uses the HTTPS protocol for secure administrator communications. With respect to the TOE implementation of HTTPS, TLS version 1.1 (RFC 4346) TLS 1.2 (RFC 5246) can be used to encrypt and authenticate administration sessions between the remote browser and TOE. The TOE supports the following ciphersuites:

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

### 7.2.4 TLS

The TLS ciphersuites mean that the keying material is determined when the session is established through a Diffie-Hellman (DH) exchange which consists of:

- Server sends 2048-bit RSA public certificate
- Server generates, signs (RSA PKCS#1) and sends DH parameters and DH public value

- Client generates and sends DH public value. The keying material is then used to encrypt/decrypt (AES128 and AES256) and authenticate (HMAC-SHA1, HMAC-SHA256) the data exchange.

When a connection is first established, the server presents the 2048-bit RSA certificate to the connecting web browser. The administrator can examine the certificate to validate the identity of the TOE and then choose to continue with the connection if the certificate conforms to the expected values. Only after the certificate has been explicitlyReview accepted as valid does the above TLS 1.1, 1.2 authentication with the administrator's web browser occur with the TOE to establish the trusted channel. After this channel is established the administrator will be presented with a warning banner which must be accepted. Next the login page is presented over this channel, where the user and password credentials can be submitted back to the TOE for verification of the administrator authentication.

When the TOE uses TLS 1.1 or 1.2 for the purposes of protecting the audit logs during transit over the network the TOE will negotiate an appropriate cipher suite based on the approved list of ciphers. Audit logs are sent in real time and for each auditable event which is to be written to the TOE, the TOE will verify that the RSA certificate present on the audit server matches the certificate which was presented when the TOE was registered with the audit server. The TOE then negotiates a suitable claimed cipher with the audit server, generates and sends the DH public value for keying and encrypts the audit message which is to be sent to the TOE. The TOE supports the following ciphersuites:

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

### 7.2.5 Cryptographic Self Tests and TOE Update Integrity

The TSF provides a cryptographic function that an administrator may use to verify the integrity of the TSF executable code. When performing an update to the TOE the image is first uploaded to the TOE by a trusted administrator using the HTTPS interface or USB media. The candidate update's digital signature is then verified by the TOE prior to beginning the installation procedure. Details on this process are provided in section 7.6.2 of the ST.

During a normal boot-up sequence the TOE administrator can see on the local console the following types of integrity and self-tests in the following order:

- Configuration file tests
- FIPS AES, SHA, 3DES and RSA tests
- Firmware integrity tests
- Entropy tests
- RNG tests

Indication of successful tests would appear as follows:

Running <test>... passed

Completion of all self-tests is indicated by:

Self-tests passed

The TOE is capable of running these tests at the request of an administrator, and periodically at an administrator-specified interval not less than once a day to demonstrate the correct operation of the

cryptographic components of the TSF. The TOE will enter into a FIPS Error Mode when failure of a self-test (integrity verification self-test, or cryptographic self-test) is detected. This mode allows the TOE to enter into a secure state. These self-tests are executed on initial start-up or at the request of an administrator. Upon successful completion of these tests an audit log will be generated by the TOE.

The TOE provides a USB interface which may be used by an authorized administrator to load private keys from a USB token. For example the 2048-bit RSA certificate used by the Network Web-Based GUI can be replaced by certificates trusted by an authorized administrator. These keys/certificates are to be placed on the USB token and the load operation can be executed via a Network CLI or Network Web-Based GUI administrator session.

### 7.2.6 Conformance to NIST SP800-56

The TOE fulfills all of the NIST SP 800-56B requirements without extensions. The TOE does not implement any functionality within this standard that is listed as “should not” and “shall not”.

Specifically the TOE claims conformance to 5.1 (Cryptographic Hash Functions), 5.2 (Message Authentication Code Algorithm), 5.3 (Random Bit Generation), 5.4 (Prime Number Generators), 5.5 (Primality Testing Methods), 5.6 (Nonces), 5.9 (Key Derivation Functions for Key Establishment Schemes), 6.1 (RSA Key Pairs - General Requirements), 6.2 (Criteria for RSA Key Pairs for Key Establishment), 6.3 (RSA Key Pair Generators), 6.4 (Assurance of Validity), 6.5 (Assurance of Private Key Possession), 6.6 (Key Confirmation), and 8 (Key Agreement Schemes). The TOE complies with RSA key pair generation according to FIPS 186-2 and FIPS 186-3 in SP 800-56B.

### 7.2.7 Key and CSP storage and zeroization

The TOE maintains a number of keys and CSPs related to its secure operation. Administrative passwords are stored in the configuration file on the flash drive of the TOE and are encoded via a hash function to ensure their confidentiality. These keys are capable of being zeroized either through a format of the flash memory or through a factory reset of the TOE.

Certificates for the purposes of HTTPS and TLS connections are maintained on the flash filesystem and are not viewable through the TOE interfaces. When these keys are no longer required the administrator can remove the keys through the formatting of the flash memory. Details on this process are contained in the FIPS level 1 security policy of the TOE.

Additionally the TOE stores a number of CSPs in volatile memory during normal operation of the cryptographic module. These CSPs include the ephemeral keys and copies of the persistent keys described above are loaded into memory during normal operation. The TOE maintains these keys in its volatile memory in order to support the TLS and HTTPS connections to the TOE. These CSPs include:

- The RSA signature generation key
- The RSA private key decryption key
- AES encryption/decryption key
- AES CMAC generation/verification key
- HMAC Key
- Diffie-Hellman Private agreement key
- RNG Seed

These CSPs are cleared when the process terminates. Each of the CSPs are protected from unauthorized access via the firmware memory management which disallows any memory reads from other processes within the OS ensuring that the CSPs are only available to the calling application.

## 7.3 User Data Protection

The TOE ensures that no information from previously processed information flows is transferred to subsequent information flows. This applies both to information that is input to the TOE from an external source and to information (e.g., padding bits) that might be added by the TOE during processing of the information from the external source. The removal of any previous residual information is done through the zeroization of data when the memory structure is initially created and strict bounds checking on the data prior to it being assigned in memory.

## 7.4 Identification and Authentication

The TOE uses a local password database for all of its credentials by default. The TOE supports local and remote users to authenticate to this local password database using a pre-shared key. Passwords can be created through the usage of mixed case characters, digits and the special characters “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, and “)”. The configuration file passwords are obscured from their ASCII form by encrypting the configuration file using an AES-128 key. This AES key is unique to the TOE and generated when the TOE enters FIPS mode and uses the hardware based entropy source and FIPS approved module to generate it.

The TOE supports only local authentication sources and no administrative action is possible prior to authentication. The TOE will consult the locally configured credential storage to ensure the provided credentials match the same one-way function as are stored in the TOE configuration files for a pre-shared key.

### 7.4.1 Web/HTTPS

By default the web/HTTPS interface is enabled on the TOE LAN ports. The TOE may also be configured to allow or disallow access to this TSFI on a per-network port basis in either the CLI or the web UI. The HTTPS web interface is accessed by going to the TOE IP on port 443. Once connected to the port and the HTTPS session is established the TOE provides a warning banner according to FTA_TAB.1 which the administrator must accept prior to proceeding. Following this banner the administrator will then be presented with a username and login screen. The administrator will then provide their credentials which are protected in transit and accepted by the webserver. Once the authentication has been received the local credential store is consulted for the pre-shared key and if the entries match access is granted to the TOE. Unsuccessful attempts to authenticate on this TSFI will be logged to the audit log. During the authentication process the user’s password is entered in a “password” input box. A failed authentication attempt will be met with the ‘’ Authentication failure. Please try again...” error message.

Successful authentication may be observed in one of two ways. The administrator may be met with a post authentication warning forcing them to accept the warning prior to proceeding. If this configuration option is not enabled the main login dashboard will be presented. By default this dashboard will contain the hostname, serial number and a number of other pieces of information regarding the TOE.

### 7.4.2 Local Console

The local console is only accessible through the use of the dedicated serial management port present on the TOE and requires that the management station be appropriately configured. For details regarding the software requirements including the console software requirements in the IT environment please see section 1.6.3

By default the local access is enabled and may not be disabled. An administrator is required to identify themselves over this interface via usage of a username text prompt. Next the pre-login warning banner is displayed as configured by the administrator. Following the banner display the user is requested to put in their password which is hidden and provides no feedback indicating any progress. Once the identification and authentication has been received the password is encrypted by the TOE's AES-128 key and compared to the local credential store. If the provided credentials match access is granted to the TOE and a CLI session is established. Next a post-login warning may be seen and the TOE will change the command prompt to the hostname followed by # and be ready to accept CLI commands from the serial console.

## 7.5 Security Management

The security management for the TOE is implemented on a per-interface basis. Regardless of the interface no management functionality is possible prior to authentication. The TOE is capable of having custom roles defined however, only the Authorized Administrator role who can administer all functionality of the TOE is defined. Administrative actions can be performed through a CLI or web-based console. Communications between administrators and the TOE are secured using TLS. Access to privileged components must be explicitly allowed or denied, including network configuration, alerts, reports, and DLP archive files.

### 7.5.1 Local Console CLI

The CLI requires identification and authentication prior to any administrative session being established with the TOE as described in section 7.4.2. Sessions are terminated after inactivity to ensure that stale sessions may not be hijacked through physical access to the serial port or through an unattended administrator workstation. Any attempt by an administrator to access the CLI without a valid session will be rejected and the administrator will be forced to authenticate.

Once authenticated to the CLI, the administrator has the ability to create and manage all TSF data locally. This includes querying the TOE version, uploading and validating an update prior to installation, installing updates, generation and maintenance of all user and administrative account sand the configuration of cryptographic module and protocols.

### 7.5.2 Web UI

The TOE tracks administrative sessions on the Web UI through the use of cookies and a session database on the TOE. When an administrator logs onto the TOE the environment supplied cookie is compared against an internally stored session database to determine if there is already an open session for this instance which has a valid session life. In the event that there is no pre-existing session established or the established session has timed out for the management of the TOE the user is redirected to the login

banner which must be accepted prior to being presented the login page. Stale administrative sessions are removed from the session database in the TOE after a period of inactivity to ensure that unattended administrator sessions can't be hijacked.

Once authenticated the Web UI gives administrators full remote control over all aspects of the TOE. This includes querying the TOE version, uploading and validating an update prior to installation, installing updates, generation and maintenance of all user and administrative accounts and the configuration of cryptographic module and protocols.

## 7.6 Protection of the TSF

The FortiAnalyzer™ appliance use a number of methods to protect themselves and the communications channels which it provides from potentially hostile entities. The TSF is able to monitor the audit events and recognize a potential security violation based on the severity of an event, or the number of events occurring within a preset time period. This includes an internal clock source provided by the kernel of the TOE which allows the auditable events to be reviewed in a reliable manner to reproduce the sequence of events that was observed. The TOE maintains its own timestamp which is used for time-sensitive operations for generating audit logs and cryptographic key regeneration intervals.

### 7.6.1 Cryptographic Key and Password Storage

The TOE itself is a FIPS 140-2 level 1 cryptographically validated module. This means that it has a number of physical security protections in place including but not limited to the protection of any keys provided to or stored within the cryptographic module. Cryptographic keys within this module are generated and destroyed per the FIPS guidelines and are not capable of being viewed through the CLI or Web interface. The TOE does not provide any method of direct access to view these keys over either of these interfaces.

Cryptographic keys related to the HTTPS GUI and audit interfaces are stored in encrypted form on the local filesystem of the TOE. An authorized administrator can generate a certificate signing request from the TOE and import the signed certificate back into the TOE for the HTTPS GUI interface. Once the certificate is imported into the TOE this information cannot be viewed again through any of the TSFI's.

Pre-shared keys related to administrator passwords and other credentials for the secure operation of the TOE are stored in the TOE's encrypted configuration file. Authorized administrators are allowed to enter configuration information through one of the protected communications paths such as the CLI or HTTPS GUI. Once the password is entered the TOE encrypts the password using AES-128 to the configuration file permanently obscuring the contents. The TOE will only display this encrypted cipher text upon a backup of the configuration file. Verification of credentials is done against this encrypted password by encrypting the passphrase entered on the TSFI and comparing it to the encrypted cipher text in the configuration file. This configuration file with the encrypted password hashes is available through downloading the configuration file using the local console or HTTPS GUI.

### 7.6.2 FortiAnalyzer™ Product Updates

The TOE protects itself during updates through the use of a cryptographic signature. The update process goes as follows. The administrator downloads the TOE to their workstation from https://support.fortinet.com. The administrator can then verify the integrity of the update by initiating the update process. To do this the administrator will then copy the file to the TOE via a trusted path such as the HTTPS GUI.

Once the firmware update is uploaded to the TOE a 2048 bit RSA signature is verified for any TOE firmware build. The signature is compared to a known key value stored on the TOE and hardcoded into the previous firmware image. Before proceeding with a firmware upgrade via the GUI or Serial CLI, the following process is followed when in the evaluated mode of operation:

- If signature is not present-> abort upgrade
- Extract public key and signature from the firmware
- Validate that public key is same as is stored previously on the TOE. If the public keys do not match abort the upgrade.
- Validate image signature using public key from the update. If the image validation using the public key fails abort upgrade.

If the firmware load test fails, the error message displayed is “File is not an update file.” Otherwise the TOE displays “upgrade successful” and reboots.

### 7.6.3 Self-Tests

The TOE performs a number of self-tests at start-up and on an ongoing basis. At start up the TOE undergoes the following tests in order:

- CPU and Memory 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 has a random pattern written and read from the memory.
- Boot loader image verification – the boot loader will compare the image of the TOE to a known checksum of the image prior to booting.
- Noise source tests – the noise source is started and pattern analysis is done on the output to ensure that the source is not stuck in a cryptographically weak state. These include both the repetition and adaptive proportion tests.
- FIPS 140-2 Known Answer Tests (KAT) – comparison of a number of cryptographic functions against an expected set of values

The TOE is also capable of performing the following tests on-demand

- FIPS 140-2 KAT (as described above)
- Noise source (as described above)

The TOE also performs the following ongoing self-tests

- Noise source pattern analysis (as described above)

The self-tests listed above provide assurance that the TOE is operating correctly by checking to ensure the CPU, BIOS, noise sources, TOE image, and FIPS 140-2 KAT are functioning as expected. Any failures would result in the TOE entering error mode and shutting down. If the tests successfully complete, the administrator has sufficient assurance to know that the hardware components are functioning as expected and the software image that has been loaded matches the appropriate digital signature.

## 7.7 TOE Access

The TOE has a number of methods to restrict access to only those administrators who are authorized to administer the TOE. The first is a login warning prior to allowing a user to log in stating that this is a restricted access system and only authorized administrators should attempt to login. This prompt is present on the local console as well as the HTTPS GUI.

The TOE also provides a method for both local and remote sessions to be protected in the event of an Administrator leaving their session unattended. An authorized administrator can configure the TOE to terminate inactive local and remote sessions following a specified period of time. By default in the evaluated configuration this timeout value is the same and it is set to 10 minutes. Finally should an administrator wish to terminate their session the TOE is able to terminate their session from the TOE side. On the local console and the HTTPS GUI the user's session is terminated and removed from the session database prior to the user being taken back to the warning banner stating that this is a restricted access system which they are forced to accept prior to going to the login page.

## 7.8 Trusted Path/Channels

The TOE is designed for secure operation by a trusted administrator using HTTPS and securing network traffic using TLS for receiving events from monitored entities.

A trusted path for an administrator to communicate with the TOE is implemented through the HTTPS GUI. When in the evaluated configuration this is the only method of remote communication which is possible with the TOE. When a remote administrator initiates a connection on this interface the TOE will respond with a cryptographically secured communication path to the workstation of the administrator which will be used for all communication between the TOE and the authorized administrator. The TOE will detect, log or reject any packets which indicate that the communications on this path have been tampered with or modified.

When the TOE is properly configured communications with the remote event-generating Fortinet devices are secured using TLS to ensure that the messages are protected in transit. The TOE will negotiate an appropriate cipher and confirm the identity of the device prior to receiving the communication. All communications are validated against the TOE identity provided during initial registration of the entity with the TOE. Should an auditable event be generated the IT entity will consult the known identity of the TOE and encrypt the audit message appropriately.

To properly configure the TOE to establish trusted channel with the other trusted FortiGate devices, the administrator can reference the CLI guide[10] and FIPS/CC supplement documents to establish trusted communication.

The TOE is capable of detecting modification or tampering of the communications on the TLS channel. In the event that a tampered or modified packet is observed on the channel the TOE will discard the packet.

[10] FortiAnalyzer 5.2.4 CLI Reference

# 8 RATIONALE

This ST claims Exact Conformance to Network Devices Protection Profile v1.1 and the NDPP Errata #3. Hence, conformance claim rationale, security objectives rationale, extended SFR rationale, and security requirements rationale (including dependency rationale, SAR choice rationale) are explicitly addressed by the Protection Profile, without further elaboration in this ST.

# 9 ACRONYMS

Table 12 – Acronyms

| Acronym | Definition |
|---|---|
| ADOM | Administrative Domain |
| CBC | Cipher Block Chaining |
| CC | Common Criteria |
| CEM | Common Evaluation Methodology |
| CLI | Command Line Interface |
| CSP | Critical Security Parameters |
| DLP | Data Leak Prevention |
| DRBG | Deterministic Random Bit Generator |
| FIPS | Federal Information Processing Standard |
| IPS | Intrusion Prevention System |
| IT | Information Technology |
| NDPP | Network Device Protection Profile |
| OSP | Organizational Security Policy |
| PP | Protection Profile |
| RAID | Redundant Array of Inexpensive Disks |
| RBG | Random Bit Generator |
| SAR | Security Assurance Requirement |
| SFR | Security Functional Requirement |
| ST | Security Target |
| TOE | Target of Evaluation |
| TSF | TOE Security Functionality |
| VDOM | Virtual Domain |

# 10 APPENDIX A –HARDWARE PLATFORM DETAILS

The following section describes a high level overview of the various claimed and supported FortiGate products examined during this evaluation.

Additional detail with regards to the various product models including performance characteristics for each model are listed on Fortinet's webpage.

The products below are listed based on their physical form factor. For additional information on the various hardware models including the number of ports and compatible expansion options please see http://www.fortinet.com/products/index.html.

## 10.1 Hardware Form Factor

| Model | CPU | interface | RAM | Flash | Storage |
|---|---|---|---|---|---|
| FAZ-200D | Intel Celeron G540 Sandy Bridge, 2.50GHz, C202 | 4 x 10/100/1000 ports | 4GB (DDR3-1333 2GBx2) | 2 GB | 1 TB |
| FAZ-1000C | Intel Xeon E5504 2.0GHz | 4 x 10/100/1000 ports | 6 GB | 2 GB | 2TB x 1 (Max 8 TB) |
| FAZ-1000D | Intel Xeon E3-1225v2 Ivy Bridge Quad Cores, 3.20GHz, C202 | 6 x 10/100/1000 RJ45 Ports, 2 x SFP Ports | 16GB (DDR3-1333 4GBx4) | 2GB | 8 TB |
| FAZ-2000B | Xeon 2.0Ghz E5504 | 6 x 10/100/1000 ports | 3 GB | None | 2 TB (Max 6 TB) |
| FAZ-3000D | Intel Xeon E5-2620, 6 Cores, 2.0GHz x 2 | 4 x 10/100/1000 RJ45 Ports, 2 x GbE SFP Ports | 16GB (DDR3 2GBx8) | 2GB | 8 X 2TB |
| FAZ-3000E | Intel Xeon E5-2620v2, 6 Cores, 2.1GHz x 2 | 4 x 10/100/1000 RJ45 Ports, 2 x GbE SFP Ports | 64GB (DDR3 4GBx16) | 2GB | 16 TB |
| FAZ-3500E | 2 x Intel Xeon E5-2630v2 6C12T 2.6GHz Sandy Bridge | 2x 10/100/1000 RJ45 ports, 2x GbE SFP ports | 64GB (8 x 8GB DDR3-1600 ECC Reg Memory) | 2GB | 24 TB (Max 48 TB) |
| FAZ-3900E | Intel Xeon E5-2630v2, 6 Cores, 2.6GHz x 2 | 2 x 10/100/1000 RJ45 Ports, 2 x | 128GB (DDR3 8Gx16) | 2 GB | 15 TB |

| | | | | | |
|---|---|---|---|---|---|
| | | 10GbE SFP+ Ports | | | |
| FAZ-4000B | Intel Westmere Quad-Core E5620 2.40GHz | 2 x 10/100/1000 ports and 2 x SFP ports | 12 GB | None | 6 TB (P24 TB Optional, 16 TB File System) |