Fortinet FortiSwitch v7.6

Security Target

Version 1.0

2025-08-07

Prepared for:

Fortinet, Inc.

899 Kifer Road
Sunnyvale, California 94086

Prepared by:

Common Criteria Testing Laboratory
6841 Benjamin Franklin Drive
Columbia, Maryland 21046

Table of Contents

1 Security Target Introduction 1
1.1 Security Target, Target of Evaluation, and Common Criteria Identification 1
1.2 Conformance Claims 4
1.3 Conventions 6
1.4 Abbreviations and Acronyms 6
1.5 Terminology 7
1.6 TOE Overview 8
1.7 TOE Description 8
1.7.1 Physical Scope 8
1.7.2 Logical Scope 10
1.8 TOE Documentation 11
1.9 Excluded Functionality 11
2 Security Problem Definition 13
3 Security Objectives 14
4 IT Security Requirements 15
4.1 Extended Requirements 15
4.2 TOE Security Functional Requirements 15
4.2.1 Security Audit (FAU) 17
4.2.2 Cryptographic Support (FCS) 20
4.2.3 Identification and Authentication (FIA) 26
4.2.4 Security Management (FMT) 28
4.2.5 Protection of the TSF (FPT) 29
4.2.6 TOE Access (FTA) 30
4.2.7 Trusted Path/Channels (FTP) 30
4.3 TOE Security Assurance Requirements 30
5 TOE Summary Specification 32
5.1 Security Audit 32
5.1.1 Audit Data Generation 32
5.1.2 Audit Storage and Audit Record Export 32
5.2 Cryptographic Support 33
5.2.1 Cryptographic Operations 34
5.2.2 Random Bit Generation 35
5.2.3 Cryptographic Key Generation and Establishment 35
5.2.4 Cryptographic Key Destruction 36
5.2.5 Cryptographic Protocols 38
5.3 Identification and Authentication 40
5.3.1 User Identification and Authentication 40
5.3.2 Authentication Failure Management 41
5.3.3 X.509 Certificate Validation 41
5.3.4 X.509 Certificate Authentication 42
5.3.5 X.509 Certificate Requests 42

5.4 Security Management 42
5.4.1 Security Roles and Specification of Management Functions 42
5.4.2 Management of Security Functions Behavior 43
5.4.3 Management of TSF Data 44
5.5 Protection of the TSF 44
5.5.1 Protection of Administrator Passwords 44
5.5.2 Protection of TSF Data (for reading of all pre-shared, symmetric and private keys) 44
5.5.3 TSF Testing 44
5.5.4 Trusted Update 45
5.5.5 Reliable Time Stamps 46
5.6 TOE Access 46
5.6.1 Access Banner 46
5.6.2 Session Termination 47
5.7 Trusted Path/Channels 47
6 Protection Profile Claims 48
7 Rationale 49
7.1 TOE Summary Specification Rationale 49

List of Figures and Tables

Table 1: Abbreviations and Acronyms 6
Table 2: Terminology 7
Table 3: Excluded Functionality 12
Table 4: Security Objectives for the Operational Environment 14
Table 5: TOE Security Functional Components 15
Table 6: Security Functional Requirements and Auditable Events 17
Table 7: Assurance Components 31
Table 8: Cryptographic Functions Implemented by OpenSSL/OpenSSH 33
Table 9: Key Clearing 36
Table 10: TSF Self-testing 44
Table 11: Security Functions vs. Requirements Mapping 49

1 Security Target Introduction

This section introduces the Target of Evaluation (TOE) and provides the Security Target (ST) and TOE references, TOE overview, and TOE description. It also contains the ST and TOE conformance claims, ST conventions, glossary, and list of abbreviations.

This ST includes the following additional sections:

- Security Problem Definition (Section 2)—describes the threats and assumptions that define the security problem to be addressed by the TOE and its environment
- Security Objectives (Section 3)—describes the security objectives for the TOE and its operational environment necessary to counter the threats and satisfy the assumptions that define the security problem
- IT Security Requirements (Section 4)—specifies the security functional requirements (SFRs) and security assurance requirements (SARs) to be met by the TOE
- TOE Summary Specification (Section 5)—describes the security functions of the TOE and how they satisfy the SFRs
- Protection Profile Claims (Section 6)—provides rationale supporting the claims for conformance of the ST and the TOE to [cPPND] and [PKG_SSH]
- Rationale (Section 7)—provides mappings and rationale for the security problem definition, security objectives, security requirements, and security functions to justify their completeness, consistency, and suitability.

1.1 Security Target, Target of Evaluation, and Common Criteria Identification

ST Title: Fortinet FortiSwitch v7.6 Security Target

ST Version: Version 1.0

ST Date: 2025-08-07

TOE Identification: Fortinet FortiSwitch v7.6 (tested version was build 8083)

The appliance models and CPUs are:

FortiSwitch v7.6 (FS) Model	Specifications	CPU
FS-T1024E	24x 1G/2.5G/5G/10GBASE-T ports and 2x 40GE / 100GE QSFP+ / QSFP28 ports RJ-45 Serial Console Port 8 GB DDR4 DRAM 32 MB NOR Flash	Intel Atom C3338 Denverton microarchitecture
FS-1024E	24x GE/10GE SFP+ ports and 2x 40GE / 100GE QSFP+ / QSFP28 ports RJ-45 Serial Console Port 8 GB DDR4 DRAM 32 MB NOR Flash	
FS-624F		

FS-624F FPOE FS-648F	24x 1GE/2.5GE/5GE RJ45 ports and 4 RJ-45 Serial Console Port Dedicated Management 10/100/1000 Port 4 GB DDR4 DRAM 32 MB Flash	
FS-648F FPOE	32x 1GE/2.5GE, 16x 1GE/2.5GE/5GE RJ45 ports and 8x 10GE/25GE SFP+/SFP28 ports RJ-45 Serial Console Port Dedicated Management 10/100/1000 Port 4 GB DDR4 DRAM 32 MB Flash	
FS-424E FIBER	Network Interfaces: 24x GE RJ45 and 4x10 GE SFP+ ports Note: SFP+ ports are compatible with 1 GE SFP Dedicated Management 10/100 Port RJ-45 Serial Console Port 1GB DDR4 DRAM 256 MB FLASH	Broadcom BCM56174 with ARM A9 processor ARM v7 microarchitecture
FS-424E	Network Interfaces: 24x GE RJ45 and 4x10 GE SFP+ ports Note: SFP+ ports are compatible with 1 GE SFP Dedicated Management 10/100 Port RJ-45 Serial Console Port 1GB DDR4 DRAM 256 MB FLASH	Broadcom BCM56160 with ARM A9 processor ARMv7 microarchitecture
FS-424E POE	Network Interfaces: 24x GE RJ45 and 4x10 GE SFP+ ports Note: SFP+ ports are compatible with 1 GE SFP Dedicated Management 10/100 Port RJ-45 Serial Console Port Power over Ethernet (PoE) Ports: 24 (802.3af/at) 1GB DDR4 DRAM 256 MB FLASH	Broadcom BCM56160 with ARM A9 processor ARMv7 microarchitecture
FS-424E FPOE	Network Interfaces: 24x GE RJ45 and 4x10 GE SFP+ ports Note: SFP+ ports are compatible with 1 GE SFP Dedicated Management 10/100 Port RJ-45 Serial Console Port Power over Ethernet (PoE) Ports:	Broadcom BCM56160 with ARM A9 processor ARMv7 microarchitecture

	24 (802.3af/at) 1GB DDR4 DRAM 256 MB FLASH	
FS-M426E FPOE	16x GE RJ45, 8x 2.5 GE RJ45 ports, 2x 5 GE RJ45, and 4x 10 GE SFP+ ports Note: SFP+ ports are compatible with 1 GE SFP RJ-45 Serial Console Port Dedicated Management 10/100 Port 1 GB DDR4 DRAM 256 MB Flash	
FSR-424F POE	12x 1/2.5 GE RJ45, 12x 1/2.5 GE SFP+ 4x 10G SFP+, 2x 40G QSFP+ ports Dedicated Management 10/100/1000 Port RJ-45 Serial Console Port 1GB DDR4 DRAM 256 MB FLASH	Broadcom BCM56170 with ARM A9 processor ARM v7 microarchitecture
FS-448E	Network Interfaces: 48x GE RJ45 and 4x 10GE SFP+ ports Note: SFP+ ports are compatible with 1 GE SFP Dedicated Management 10/100 Port RJ-45 Serial Console Port 1GB DDR4 DRAM 256 MB FLASH	Broadcom BCM56172 with ARM A9 processor ARMv7 microarchitecture
FS-448E POE	Network Interfaces: 48x GE RJ45 and 4x 10GE SFP+ ports Note: SFP+ ports are compatible with 1 GE SFP Dedicated Management 10/100 Port RJ-45 Serial Console Port Power over Ethernet (PoE) Ports: 48 (802.3af/at) 1GB DDR4 DRAM 256 MB FLASH	
FS-448E FPOE	Network Interfaces: 48x GE RJ45 and 4x 10GE SFP+ ports Note: SFP+ ports are compatible with 1 GE SFP Dedicated Management 10/100 Port RJ-45 Serial Console Port Power over Ethernet (PoE) Ports: 48 (802.3af/at) 1GB DDR4 DRAM 256 MB FLASH	

FS-1048E	48x GE/10 GE SFP+ ports and 6x 40 GE QSFP+ ports or 4x 100 GE QSFP28 port RJ-45 Serial Console Port 8GB DDR3 128MB NOR	Intel Atom C2538 Rangeley microarchitecture
FS-3032E	32x 40 GE / 100 GE QSFP+ / QSFP28 ports RJ-45 Serial Console Port 8GB DDR3 128MB NOR	
FS-2048F	48x 1GE/10GE/25GE SFP28 ports 2x 1GE/10GE SFP+ ports 8x 40GE / 100GE QSFP28 ports RJ-45 Serial Console Port 8GB DDR4 8GB NAND	Intel Atom C3558 Denverton microarchitecture

TOE Developer: Fortinet, Inc.

Evaluation Sponsor: Fortinet, Inc.

CC Identification: Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 5, April 2017.

1.2 Conformance Claims

This ST and the TOE it describes are conformant to the following CC specifications:

- Common Criteria for Information Technology Security Evaluation Part 2: Security Functional Components, Version 3.1 Revision 5, April 2017
  - Part 2 Extended
- Common Criteria for Information Technology Security Evaluation Part 3: Security Assurance Components, Version 3.1 Revision 5, April 2017
  - Part 3 Conformant.

This ST and the TOE it describes are conformant to the following:

- *collaborative Protection Profile for Network Devices*, Version 3.0e, 06 December 2023 [cPPND], including the following optional and selection-based SFRs: FAU_STG.1, FAU_STG_EXT.3, FCS_HTTPS_EXT.1; FCS_TLSC_EXT.1; FCS_TLSS_EXT.1; FIA_AFL.1; FIA_PMG_EXT.1; FIA_UAU.7; FIA_X509_EXT.1/Rev; FIA_X509_EXT.2; FIA_X509_EXT.3; FMT_MTD.1/CryptoKeys; FPT_APW_EXT.1; and FTA_SSL_EXT.1.
- Functional Package for Secure Shell (SSH), Version 1.0, 13 May 2021 [PKG_SSH], including the following selection-based SFRs: FCS_SSHS_EXT.1.

The following NIAP Technical Decisions are applicable to the claimed Protection Profile/Package:

- TD0682 – Addressing Ambiguity in FCS_SSHS_EXT.1 Tests
  - This TD archives TD0666 and is applicable to the TOE but relates solely to evaluation activities so it does not affect the ST.

- TD0695– Choice of 128 or 256 bit size in AES-CTR in SSH Functional Package.
    - This TD modifies an explanation in the PP but does not change any requirements.
- TD0732 – FCS_SSH_EXT.1.3 Inconsistency
    - This TD archives TD0694 and is applicable to the TOE but relates solely to the evaluation activities so it does not affect the ST.
- TD0777 – Clarification to Selections for Auditable Events for FCS_SSH_EXT.1
    - This TD is applicable to the TOE.
- TD0836 – NIT Technical Decision: Redundant Requirements in FPT_TST_EXT.1
    - This TD is applicable to the TOE.
- TD0868 – NIT Technical Decision: Clarification of time frames in FCS_IPSEC_EXT.1.7 and FCS_IPSEC_EXT.1.8
    - This TD is not applicable to the TOE, since the SFR it applies to has not been claimed.
- TD0879 – NIT Decision: Correction of Chapter Headings in CPP_ND_V3.0E
    - This TD corrects an administrative error in the PP and does not directly affect the ST or the TOE.
- TD0880 – NIT Decision: Removal of Duplicate Selection in FMT_SMF.1.1
    - This TD is applicable to the TOE.
- TD0886 – Clarification to FAU_STG_EXT.1 Test 6
    - This TD is applicable to the TOE but relates solely to the evaluation activities so it does not affect the ST.
- TD0899 – NIT Technical Decision: Correction of Renegotiation Test for TLS 1.2
    - This TD is applicable to the TOE but relates solely to the evaluation activities so it does not affect the ST.
- TD0900 – NIT Technical Decision: Clarification to Local Administrator Access in FIA_UIA_EXT.1.3
    - This TD is applicable to the TOE but modifies an SFR selection in FIA_UIA_EXT.1.3 that the TSF does not claim, so there is no change made to the ST as a result of applying this TD.
- TD0909 – Updates to FCS_SSH_EXT.1.1 App Note in SSH FP 1.0
    - This TD is applicable to the TOE but modifies an application note in FCS_SSH_EXT.1.1 so it does not affect the ST.
- TD0921 – NIT Technical Decision: Addition of FIPS PUB 186-5 and Correction of Assignment
    - This TD is applicable to the TOE.
- TD0923 – NIT Technical Decision: Auditable event for FAU_STG_EXT.1 in FAU_GEN.1.2
    - This TD is applicable to the TOE but relates solely to the application note so it does not affect the ST.

1.3 Conventions

The following conventions are used in this document:

- Security Functional Requirements—Part 1 of the CC defines the approved set of operations that may be applied to functional requirements: iteration; selection; assignment; and refinement.
  - Iteration—allows a component to be used more than once with varying operations. In this ST, the only iterated requirements are those reproduced from [cPPND], which uses descriptive strings to distinguish iterations of a requirement. For example, iterations of FCS_COP.1 are identified FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, and FCS_COP.1/KeyedHash.
  - Selection—allows the specification of one or more elements from a list. Selections completed in the ST are indicated using bold italics and are enclosed by brackets (e.g., [***selection***]).
  - Assignment—allows the specification of an identified parameter. Assignments completed in the ST are indicated using bold text and are enclosed by brackets (e.g., [**assignment**]). An assignment within a selection is identified in bold italics and with embedded bold brackets (e.g., [***[selected-assignment]***]).
  - Refinement—allows the addition of details. Refinements made in the ST of requirements drawn from [cPPND] would be indicated using bold for additions and strike-through for deletions (e.g., "… ~~some~~ **all** objects).
- Other sections of the ST—other sections of the ST use bolding and/or different fonts (such as `Courier`) to highlight text of special interest, such as captions, commands, or filenames specific to the TOE.

1.4 Abbreviations and Acronyms

Table 1: Abbreviations and Acronyms

Abbreviation	Definition
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CLI	Command Line Interface
DH	Diffie-Hellman
DRBG	Deterministic Random Bit Generator
DSA	Digital Signature Algorithm
ECDSA	Elliptic Curve Digital Signature Algorithm
FS	Shorthand for the TOE: Fortinet FortiSwitch
GUI	Graphical User Interface
HMAC	Hash-based Message Authentication Code
LAN(s)	Local Area Networks
MAC	Message Authentication Code
NSS	Network Security Services
NTP	Network Time Protocol
SHA	Secure Hash Algorithm

Abbreviation	Definition
SSH	Secure Shell
TLS	Transport Layer Security
TOE	Target of Evaluation
TSF	TOE Security Functionality

1.5 Terminology

Table 2: Terminology

Term	Definition
SD-branch	A single, automated, centrally managed software-centric platform that replaces or supplements an existing branch network architecture.

1.6 TOE Overview

The Fortinet FortiSwitch v7.6 TOE is a series of scalable Ethernet Switch network devices used to connect devices on Ethernet local area networks (LANs). The TOE is optimal for use in converged network environments; enabling voice, data, and wireless traffic to be delivered across a single network.

The focus of this evaluation is on the TOE functionality supporting the claims in the collaborative Protection Profile for Network Devices [cPPND] and Functional Package for Secure Shell (SSH) [PKG_SSH]. The security functionality specified in [cPPND] includes protection of communications between the TOE and external IT entities, identification and authentication of administrators, auditing of security-relevant events, ability to verify the source and integrity of updates to the TOE, and use of NIST-validated cryptographic mechanisms. [PKG_SSH] specifies the security requirements for the SSH Protocol. The TOE focuses on the security of the network channels used for syslog, management, and authentication. Therefore, the transfer of voice data and wireless traffic was not evaluated.

1.7 TOE Description

The TOE is a series of scalable Ethernet Switch network devices used to connect devices on Ethernet LANs. The switches are enterprise level switches suitable for SD-Branch deployments with high throughput requirements. The TOE supports management via graphical user interface (GUI) and command line interface (CLI). The TOE must be configured into and operated in FIPS-CC mode.

For the purpose of this evaluation, the TOE is treated as a network device offering NIST validated cryptographic functions, security auditing, secure administration, trusted updates, self-tests, and secure connections to other servers (i.e., to export audit records), protected using HTTPS, TLS and SSH.

Cryptographic functionality is performed by the FortiSwitch Crypto Library v7.6 included in the TOE in support of higher-level protocols (TLS, SSH). The module's FIPS-Approved cryptographic algorithms have obtained CAVP certificates.

The TOE audits security relevant events, stores audit records locally, and can be configured to forward its audit records to an external syslog server in the network environment. An administrator can manually set the TOE's time.

The TOE uses TLS to protect syslog, offers a management GUI protected by TLS/HTTPS; and provides a management CLI protected by SSH.

Administrators are able to query the current version of the product firmware and manage the security functions of the TOE including perform updates on the product. Public/private keys are used to provide digital signatures for protection of the update files.

The TOE provides self-tests to ensure the integrity and correct operation of the TOE.

1.7.1 Physical Scope

The TOE is deployed as a single physical appliance in a network. The models included in the TOE are identified in Section 1.1 and include the v7.6 software.

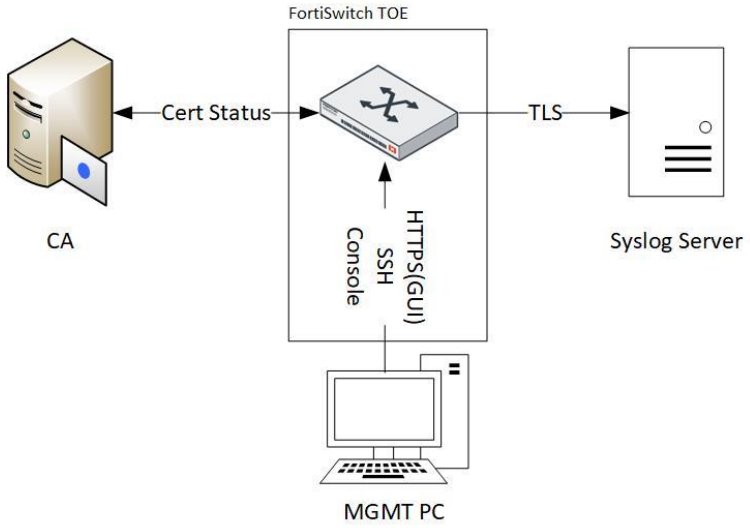
The models differ in terms of number and types of network interfaces. All devices have an RJ-45 serial console port and a dedicated management port. Both provide access to the management interfaces. See Section 1.1 for specification details. All security-relevant functionality is identical across all firmware images. There are no security relevant differences between the appliance models.

Each TOE appliance includes one Intel Atom or Broadcom processor implementing the microarchitecture identified in Section 1.1. The TOE appliances include Fortinet-developed firmware and the FortiSwitchOS v7.6 operating system, developed by Fortinet that is based on the Linux kernel. The operating system has been hardened and does not permit operators (even an authorized administrator) access to the OS. Fortinet has provided a list of included third party components separately, for the purposes of the evaluation activities.

The TOE offers both CLI and GUI management interfaces. The administrator gains access to the CLI locally by using the supplied console cable to connect the device's console port to the serial port on the management computer and using a terminal program. All devices also provide a dedicated management port that can be connected to locally using an Ethernet cable connected directly to the management computer, or remotely by connecting to a management network. Whether connected locally or remotely, the administrator can access both the GUI (using HTTPS) and the CLI (using SSH) via the dedicated management port. All administrator authentication is performed by the TOE, using either username/password (for GUI and CLI access) or username/public key (for CLI access).

Figure 1 shows the TOE management interfaces as well as the interface with the remote audit server.

Figure 1 - TOE Interfaces



The TOE in its evaluated configuration requires the following components in its operational environment:

- A TLS-protected syslog server that receives audit events from the TOE
- A Certificate Authority for validation of X.509 certificates
- A client workstation for administrator access to the web GUI and CLI with:
  - A supported browser:
    - Microsoft Edge, version 112
    - Mozilla Firefox®, version 113
    - Google Chrome™ web, version 113.
  - An SSH Client for remote access to the CLI.

1.7.2 Logical Scope

This section summarizes the security functions provided by the TOE:

- Security audit
- Cryptographic support
- Identification and authentication
- Security management
- Protection of the TSF
- TOE access
- Trusted path/channels.

1.7.2.1 Security Audit

The TOE generates audit events associated with identification and authentication, management, updates, and user sessions. The TOE can store the events in a local log and export them to a syslog server using a TLS protected channel.

1.7.2.2 Cryptographic Support

The TOE provides CAVP certified cryptography in support of its SSH, TLS, self-testing, and for verifying TOE update package signatures. Cryptographic services include key management, random bit generation, symmetric encryption and decryption, digital signature, and secure hashing.

1.7.2.3 Identification and Authentication

The TOE requires users to be identified and authenticated before they can use functions mediated by the TOE, with the exception of viewing and agreeing to the login banner. The TOE authenticates a user's credentials (password, public key) using local mechanisms provided by the TOE. The TOE also provides X.509 certificate checking for its TLS connections.

1.7.2.4 Security Management

The TOE provides CLI and web-based management interfaces that an administrator can access locally or remotely via a network port. Remote connections to the management interfaces are protected with SSH for the CLI and HTTPS for the GUI. The local administrative interface is subject to physical protection. To access the TOE locally, an administrator must directly connect their workstation to the TOE using a serial cable and successfully log in. The management interface is limited to the authorized administrators.

1.7.2.5 Protection of the TSF

The TOE implements various self-protection mechanisms. The TOE performs self-tests that cover the correct operation of the TSF. It provides functions necessary to securely update the TOE. It relies upon administrator manually provided time to ensure reliable timestamps. It protects sensitive data such as passwords and cryptographic keys stored on the TOE so that they are not accessible, even by an authorized administrator.

1.7.2.6 TOE Access

The TOE will terminate local and remote interactive sessions after a configurable period of inactivity. The TOE additionally provides the capability for administrators to terminate their own interactive sessions.

The TOE can be configured to display an advisory and consent warning message before establishing a user session.

1.7.2.7 Trusted Path/Channels

When accessed remotely, the CLI and GUI management interfaces are protected by SSH or TLS, thus ensuring protection against modification and disclosure.

The TOE protects communications with the external syslog server from modification and disclosure by using TLS.

1.8 TOE Documentation

The TOE is supplied with the following guidance documentation that describes the installation process for the TOE and provides guidance for configuration and secure use of its security features:

- NDcPP v3.0e FortiSwitch Technote – FortiSwitch 7.6, Version 1.0, August 7, 2025
- Fortinet Administration Guide—Standalone Mode FortiSwitchOS 7.6.0, October 14, 2024
- Fortinet CLI Reference—FortiSwitchOS 7.6.0, August 27, 2024
- Fortinet Log Reference FortiSwitchOS 7.6.0, August 12, 2024

Fortinet provides the following QuickStart Guides ([QSGs]) for each of the appliances within the scope of evaluation:

- FortiSwitch 1024E Series (FS-1024E, FS-T1024E) QuickStart Guide, May 2, 2022
- FortiSwitch 624F/648F Series (FS-624F, FS-624F-FPOE, FS-648F, FS-648F-FPOE) QuickStart Guide, August 21, 2024
- FortiSwitch 424E Series (FS-424E, FS-424E-Fiber, FS-424E-POE, FS-424E-FPOE) QuickStart Guide, October 7, 2022
- FortiSwitch M426E-FPOE QuickStart Guide, September 8, 2023
- FortiSwitch Rugged 424F POE (FSR-424F-POE) QuickStart Guide, May 12, 2023
- FortiSwitch 448E Series (FS-448E, FS-448E-PoE, FS-448E FPoE) QuickStart Guide, December 22, 2022
- FortiSwitch 1048E QuickStart Guide, November 18, 2021
- FortiSwitch 3032E QuickStart Guide, March 27, 2024
- QuickStart Guide FortiSwitch 2048F (FS-2048F), November 21, 2023.

1.9 Excluded Functionality

The list below identifies features or protocols that are not evaluated or must be disabled, and the rationale why. Note that this does not mean the features cannot be used in the evaluated configuration (unless explicitly stated so). It means that the features were not evaluated and/or validated by an independent third party and the functional correctness of the implementation is vendor assertion. Evaluated functionality is scoped exclusively to the security functional requirements specified in this Security Target. In particular, only the following protocols implemented by the TOE have been tested, and only to the extent specified by the security functional requirements: TLS, HTTPS, SSH. The features below are out of scope.

Table 3: Excluded Functionality

Feature	Description
HTTP Management Protocols	Only SSH and HTTPS are used for the remote management protocols to manage the TOE.
802.1X	802.1X is used for network access. The NDcPP does not define requirements for this type of traffic/protocol and therefore it has not been evaluated.
External Authentication (LDAP, RADIUS, TACACS+)	Only local authentication is used for administrative authentication.
Administrator Authentication using X.509 certificates	In the evaluated configuration, identification and authentication of administrators is restricted to password and public-key for the CLI
“Remote Management” Operation Mode	Management modes other than standalone mode (also referred to as “Local Management”) are excluded from the evaluated configuration.
FortiSwitch Cloud management	The TOE is managed in standalone mode without FortiSwitch Cloud (must not be enabled).
ICMP	By default, the TOE disables ICMP Echo Response when placed into FIPS-CC mode.
Any features not associated with SFRs in claimed [NDcPP]	NDcPP forbids adding additional requirements to the Security Target (ST). If additional functionalities or products are mentioned in the ST, it is for completeness only.

2 Security Problem Definition

This ST includes by reference the Security Problem Definition (comprising threat statements, assumptions, and organizational security policies) from [cPPND]. A.COMPONENTS_RUNNING does not apply since the TOE is not distributed. A.VS_TRUSTED_ADMINISTRATOR, A.VS_REGULAR_UPDATES, A.VS_ISOLATON, and A.VS_CORRECT_CONFIGURATION do not apply since the TOE is not virtual.

The PP offers additional information about the threats, assumptions, and organizational security policies, but that has not been reproduced here and the PP should be consulted if there is interest in that material.

In general, the [cPPND] has presented a Security Problem Definition appropriate for network infrastructure devices, and as such is applicable to the TOE.

3 Security Objectives

The [cPPND] defines the following security objectives for the operational environment of the TOE. OE.COMPONENTS_RUNNING is not applicable since the TOE is not distributed. OE.VM_CONFIGURATION is not applicable since the TOE is not virtual.

Table 4: Security Objectives for the Operational Environment

Objective	Description
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner. For TOEs supporting X.509v3 certificate-based authentication, the Security Administrator(s) are assumed to monitor the revocation status of all certificates in the TOE's trust store and to remove any certificate from the TOE's trust store in case such certificate can no longer be trusted.
OE.UPDATES	The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.
OE.RESIDUAL_INFORMATION	The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

4 IT Security Requirements

This section defines the Security Functional Requirements (SFRs) and Security Assurance Requirements (SARs) that serve to represent the security functional claims for the TOE and to scope the evaluation effort.

The SFRs have all been drawn from [cPPND] and [PKG_SSH]. As such, operations on SFRs already performed in that PP are not identified here. Rather, the SFRs have been copied from [cPPND] and [PKG_SSH] and any formatting used in the PP/Package have been removed. Operations performed on SFRs in the writing of this ST are identified in accordance with the conventions described in Section 1.3.

The SARs are the set of SARs specified in [cPPND].

4.1 Extended Requirements

All of the extended requirements in this ST have been drawn from [cPPND] and [PKG_SSH]. The [cPPND] and [PKG_SSH] define the following extended SFRs and since they are not redefined in this ST, the [cPPND] should be consulted for more information in regards to these CC extensions.

- FAU_STG_EXT.1 – Protected Audit Event Storage
- FAU_STG_EXT.3 – Action in Case of Possible Audit Data Loss
- FCS_HTTPS_EXT.1 – HTTPS Protocol
- FCS_RBG_EXT.1 – Random Bit Generation
- FCS_SSH_EXT. 1 – SSH Protocol [PKG_SSH]
- FCS_SSHS_EXT.1 – SSH Protocol – Server [PKG_SSH]
- FCS_TLSC_EXT.1 – TLS Client Protocol
- FCS_TLSS_EXT.1 – TLS Server Protocol
- FIA_PMG_EXT.1 – Password Management
- FIA_UIA_EXT.1 – User Identification and Authentication
- FIA_X509_EXT.1 – X.509 Certificate Validation
- FIA_X509_EXT.2 – X.509 Certificate Authentication
- FIA_X509_EXT.3 – X.509 Certificate Requests
- FPT_APW_EXT.1 – Protection of Administrator Passwords
- FPT_SKP_EXT.1 – Protection of TSF Data
- FPT_STM_EXT.1 – Reliable Time Stamps
- FPT_TST_EXT.1 – TSF Testing
- FPT_TUD_EXT.1 – Trusted Update
- FTA_SSL_EXT.1 – TSF-Initiated Session Locking

4.2 TOE Security Functional Requirements

The following table identifies the SFRs that are satisfied by the TOE.

Table 5: TOE Security Functional Components

Requirement Class	Requirement Component
FAU: Security audit	FAU_GEN.1 – Audit Data Generation
	FAU_GEN.2 – User Identity Association
	FAU_STG.1 – Protected Audit Trail Storage
	FAU_STG_EXT.1 – Protected Audit Event Storage

Requirement Class	Requirement Component
	FAU_STG_EXT.3 – Action in Case of Possible Audit Data Loss
FCS: Cryptographic support	FCS_CKM.1 – Cryptographic Key Generation
	FCS_CKM.2 – Cryptographic Key Establishment
	FCS_CKM.4 – Cryptographic Key Destruction
	FCS_COP.1/DataEncryption – Cryptographic Operation (AES Data Encryption/Decryption)
	FCS_COP.1/SigGen – Cryptographic Operation (Signature Generation and Verification)
	FCS_COP.1/Hash – Cryptographic Operation (Hash Algorithm)
	FCS_COP.1/KeyedHash – Cryptographic Operation (Keyed Hash Algorithm)
	FCS_HTTPS_EXT.1 – HTTPS Protocol
	FCS_RBG_EXT.1 – Random Bit Generation
	FCS_SSH_EXT.1– SSH Protocol
	FCS_SSHS_EXT.1 – SSH Protocol - Server
	FCS_TLSC_EXT.1 – TLS Client Protocol
	FCS_TLSS_EXT.1 – TLS Server Protocol
FIA: Identification and authentication	FIA_AFL.1 – Authentication Failure Management
	FIA_PMG_EXT.1 – Password Management
	FIA_UAU.7 – Protected Authentication Feedback
	FIA_UIA_EXT.1 – User Identification and Authentication
	FIA_X509_EXT.1/Rev – X.509 Certificate Validation
	FIA_X509_EXT.2 – X.509 Certificate Authentication
	FIA_X509_EXT.3 – X.509 Certificate Requests
FMT: Security Management	FMT_MOF.1/ManualUpdate – Management of Security Functions Behavior
	FMT_MTD.1/CoreData – Management of TSF Data
	FMT_MTD.1/CryptoKeys – Management of TSF Data
	FMT_SMF.1 – Specification of Management Functions
	FMT_SMR.2 – Restrictions on Security Roles
FPT: Protection of the TSF	FPT_APW_EXT.1 – Protection of Administrator Passwords
	FPT_SKP_EXT.1 – Protection of TSF Data (for reading of all pre-shared, symmetric and private keys)
	FPT_STM_EXT.1 – Reliable Time Stamps
	FPT_TST_EXT.1 – TSF Testing
	FPT_TUD_EXT.1 – Trusted update
FTA: TOE access	FTA_SSL_EXT.1 – TSF-Initiated Session Locking
	FTA_SSL.3 – TSF-Initiated Termination
	FTA_SSL.4 – User-Initiated Termination

Requirement Class	Requirement Component
	FTA_TAB.1 – Default TOE Access Banners
FTP: Trusted path/channels	FTP_ITC.1 – Inter-TSF Trusted Channel
	FTP_TRP.1/Admin – Trusted Path

4.2.1 Security Audit (FAU)

4.2.1.1 Audit Data Generation (FAU_GEN.1)

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) All administrative actions comprising:
- Administrative login and logout (name of Administrator account shall be logged if individual accounts are required for Administrators).
- Changes to TSF data related to configuration changes (in addition to the information that a change occurred it shall be logged what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference shall be logged).
- [***Resetting passwords (name of related Administrator account shall be logged)***];

d) Specifically defined auditable events listed in **Table 6**.

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, information specified in column three of **Table 6**.

Table 6: Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG.1	None.	None.
FAU_STG_EXT.1	Configuration of local audit settings.	Identity of account making changes to the audit configuration.
FAU_STG_EXT.3	Low storage space for audit events	None.
FCS_CKM.1	None.	None.
FCS_CKM.2	None.	None.
FCS_CKM.4	None.	None.
FCS_COP.1/DataEncryption	None.	None.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session	Reason for failure
FCS_RBG_EXT.1	None.	None.
FCS_SSH_EXT.1	• ***[Failure to establish a SSH session]*** • ***[Establishment of SSH connection]*** • ***[Termination of SSH connection session]*** • ***[Dropping of packet(s) outside defined size limits]***	• ***[Reason for failure and [1]Non-TOE endpoint of connection (IP Address])*** • ***[Non-TOE endpoint of connection (IP Address)]*** • ***[Non-TOE endpoint of connection (IP Address)]*** • ***[Packet size]***
FCS_TLSC_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanisms.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FIA_X509_EXT.1/Rev	• Unsuccessful attempt to validate a certificate • Any addition, replacement or removal of trust anchors in the TOE's trust store	• Reason for failure of certificate validation • Identification of certificates added, replaced or removed as trust anchor in the TOE's trust store
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update.	None.
FMT_MTD.1/CoreData	None.	None.
FMT_MTD.1/CryptoKeys	None.	None.
FMT_SMF.1	All management activities of TSF data.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.

[1] Modified by TD0777

Requirement	Auditable Events	Additional Audit Record Contents
FPT_STM_EXT.1	Discontinuous changes to time – either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1).	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session lock.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	• Initiation of the trusted channel. • Termination of the trusted channel. • Failure of the trusted channel functions.	• None • None • Reason for failure.
FTP_TRP.1/Admin	• Initiation of the trusted path. • Termination of the trusted path. • Failure of the trusted path functions.	• None • None • Reason for failure

4.2.1.2 User Identity Association (FAU_GEN.2)

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.

4.2.1.3 Protected Audit Trail Storage (FAU_STG.1)

FAU_STG.1.1 The TSF shall protect the stored audit records in the audit trail from unauthorized deletion.

FAU_STG.1.2 The TSF shall be able to prevent unauthorized modifications to the stored audit records in the audit trail.

4.2.1.4 Protected Audit Event Storage (FAU_STG_EXT.1)

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

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

- ***The TOE shall consist of a single standalone component that stores audit data locally***

].

FAU_STG_EXT.1.3 The TSF shall maintain a [***log file***] of audit records in the event that an interruption of communication with the remote audit server occurs.

FAU_STG_EXT.1.4 The TSF shall be able to store [***non-persistent***] audit records locally with a minimum storage size of [***64 KB***].

FAU_STG_EXT.1.5 The TSF shall [***overwrite previous audit records according to the following rule: [overwrite the oldest log record first]***] when the local storage space for audit data is full.

FAU_STG_EXT.1.6 The TSF shall provide the following mechanisms for administrative access to locally stored audit records [***ability to view locally***].

4.2.1.5 Action in Case of Possible Audit Data Loss (FAU_STG_EXT.3)

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

4.2.2 Cryptographic Support (FCS)

4.2.2.1 Cryptographic Key Generation (FCS_CKM.1)

FCS_CKM.1.1[2] The TSF shall generate asymmetric cryptographic keys in accordance with a specified cryptographic key generation algorithm: [

- ***RSA schemes using cryptographic key sizes of [2048-bits, 3072 bits, 4096 bits] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", A.1;***
- ***ECC schemes using 'NIST curves' [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4, or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2, or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.;***
- ***FFC Schemes using 'safe-prime' groups that meet the following: "NIST Special Publication 800-56A Revision 3, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography" and [RFC 3526]***

].

4.2.2.2 Cryptographic Key Establishment (FCS_CKM.2)

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

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

].

[2] Modified by TD0921

4.2.2.3 Cryptographic Key Destruction (FCS_CKM.4)

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

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

that meets the following: No Standard.

Application Note: *The SSH host key is overwritten with a new value if the command to regenerate that key is used, stored keys are overwritten with zeroes in all other cases (including the SSH host key if the command to destroy keys is executed).*

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

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

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

FCS_COP.1.1/SigGen[3] The TSF shall perform cryptographic signature services (generation and verification) in accordance with a specified cryptographic algorithm [

- ***RSA Digital Signature Algorithm,***
- ***Elliptic Curve Digital Signature Algorithm***

]

and cryptographic key sizes [

- ***For RSA: [modulus 2048, 3072, 4096 bits]***
- ***For ECDSA: [256, 384, 521 bits]***

],

that meet the following: [

- ***For RSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 or FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4 using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3***
- ***For ECDSA schemes implementing [P-256, P-384, P-521] curves that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST Recommended" curves;***

[3] Modified by TD0921

or FIPS_PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and NIST SP 800-186 Section 3.2.1, Implementing Weierstrass curves; or ISO/IEC 14888-3, "IT Security techniques - Digital signatures with appendix - Part 3: Discrete logarithm based mechanisms", Section 6.6.

].

4.2.2.6 Cryptographic Operation (Hash Algorithm) (FCS_COP.1/Hash)

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

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

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

4.2.2.8 HTTPS Protocol (FCS_HTTPS_EXT.1)

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

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

4.2.2.9 Random Bit Generation (FCS_RBG_EXT.1)

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

FCS_RBG_EXT.1.2 The deterministic RBG shall be seeded by at least one entropy source that accumulates entropy from [***[1] software-based noise source***] with a minimum of [***256 bits***] of entropy at least equal to the greatest security strength, according to ISO/IEC 18031:2011 Table C.1 “Security Strength Table for Hash Functions”, of the keys and hashes that it will generate.

4.2.2.10 SSH Protocol (FCS_SSH_EXT.1)

FCS_SSH_EXT.1.1 The TOE shall implement SSH acting as a [***server***] in accordance with that complies with RFCs 4251, 4252, 4253, 4254, [***4256, 4344, 5647, 5656, 6668, 8268***] and [***no other standard***].

FCS_SSH_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods: [

- ***“password” (RFC 4252),***
- ***“keyboard-interactive” (RFC 4256),***
- ***“public key” (RFC 4252): [***
  - ***ecdsa-sha2-nistp521 (RFC 5656)]***

] and no other methods.

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

FCS_SSH_EXT.1.4 The TSF shall protect data in transit from unauthorised disclosure using the following mechanisms: [

- ***aes128-ctr (RFC 4344),***
- ***aes256-ctr (RFC 4344),***
- ***aes128-gcm@openssh.com (RFC 5647),***
- ***aes256-gcm@openssh.com* (*RFC 5647)***

] and no other mechanisms.

FCS_SSH_EXT.1.5 The TSF shall protect data in transit from modification, deletion, and insertion using: [

- ***hmac-sha2-256 (RFC 6668),***
- ***hmac-sha2-512 (RFC 6668),***
- ***Implicit***

] and no other mechanisms.

FCS_SSH_EXT.1.6 The TSF shall establish a shared secret with its peer using: [

- ***ecdh-sha2-nistp256 (RFC 5656),***
- ***ecdh-sha2-nistp384 (RFC 5656),***
- ***ecdh-sha2-nistp521 (RFC 5656),***
- ***diffie-hellman-group14-sha256 (RFC 8268)***
- ***diffie-hellman-group16-sha512 (RFC 8268)***
- ***diffie-hellman-group18-sha512 (RFC 8268)***

] and no other mechanisms.

FCS_SSH_EXT.1.7 The TSF shall use SSH KDF as defined in [

- ***RFC 5656 (Section 4)***

] to derive the following cryptographic keys from a shared secret: session keys.

FCS_SSH_EXT.1.8 The TSF shall ensure that [

- ***a rekey of the session keys***

] occurs when any of the following thresholds are met:

- one hour connection time,
- no more than one gigabyte of transmitted data, or
- no more than one gigabyte of received data.

4.2.2.11 SSH Protocol - Server (FCS_SSHS_EXT.1)

FCS_SSHS_EXT.1.1 The TSF shall authenticate itself to its peer (SSH Client) using: [

- ***ecdsa-sha2-nistp521 (RFC 5656)***

].

4.2.2.12TLS Client Protocol (FCS_TLSC_EXT.1)

FCS_TLSC_EXT.1.1 The TSF shall implement [***TLS 1.2 (RFC 5246)***] supporting the following ciphersuites:

[

- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 8422***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 8422***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***

] and no other ciphersuites.

FCS_TLSC_EXT.1.2 The TSF shall verify that the presented identifier matches [***the reference identifier per RFC 6125 section 6, IPv4 address in the CN or in the SAN, IPv6 address in the CN or in the SAN***].

FCS_TLSC_EXT.1.3 The TSF shall not establish a trusted channel if the server certificate is invalid [

- ***without any administrator override mechanism.***

].

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

FCS_TLSC_EXT.1.5 The TSF shall [

- ***present the signature_algorithms extension with support for the following algorithms: [***
  - ***ecdsa_secp256r1 with sha256(0x0403),***
  - ***ecdsa_secp384r1 with sha384(0x0503),***
  - ***ecdsa_secp521r1 with sha512(0x0603),***
  - ***rsa_pkcs1 with sha256(0x0401),***
  - ***rsa_pkcs1 with sha384(0x0501),***
  - ***rsa_pkcs1 with sha512(0x0601),***
  - ***rsa_pss_rsae with sha256(0x0804),***
  - ***rsa_pss_rsae with sha384(0x0805),***
  - ***rsa_pss_rsae with sha512(0x0806),***
  - ***rsa_pss_pss with sha256(0x0809),***
  - ***rsa_pss_pss with sha384(0x080a),***
  - ***rsa_pss_pss with sha512(0x080b)***

)

] and no other algorithms;

].

FCS_TLSC_EXT.1.6 The TSF [***does not provide***] the ability to configure the list of supported ciphersuites as defined in FCS_TLSC_EXT.1.1.

FCS_TLSC_EXT.1.7 The TSF shall prohibit the use of the following extensions:
- Early data extension
- Post-handshake client authentication according to RFC 8446, Section 4.2.6.

FCS_TLSC_EXT.1.8 The TSF shall [***not use PSKs***].

FCS_TLSC_EXT.1.9 The TSF shall [***reject [TLS 1.2] renegotiation attempts***].

4.2.2.13 TLS Server Protocol (FCS_TLSS_EXT.1)

FCS_TLSS_EXT.1.1 The TSF shall implement [***TLS 1.2 (RFC 5246)***] and reject all other TLS and SSL versions. The TLS implementation will support the following ciphersuites: [
- ***TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 8422***
- ***TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 8422***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422***
- ***TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289***
- ***TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289***]

and no other ciphersuites.

FCS_TLSS_EXT.1.2 The TSF shall authenticate itself using X.509 certificate(s) using [***RSA with key size [2048, 3072, 4096] bits; ECDSA over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves***].

FCS_TLSS_EXT.1.3 The TSF shall perform key exchange using: [
- ***EC Diffie-Hellman Key agreement over NIST curves [secp256r1, secp384r1, secp521r1] and no other curves***

].

FCS_TLSS_EXT.1.4 The TSF shall support [***session resumption based on session tickets according to RFC 5077 (TLS 1.2)***].

FCS_TLSS_EXT.1.5 The TSF [***does not provide***] the ability to configure the list of supported ciphersuites as defined in FCS_TLSS_EXT.1.1

FCS_TLSS_EXT.1.6 The TSF shall prohibit the use of the following extensions:
- Early data extension

FCS_TLSS_EXT.1.7 The TSF shall [***not use PSKs***].

FCS_TLSS_EXT.1.8 The TSF shall [***support secure renegotiation in accordance with RFC 5746 by always including the "renegotiation_info" TLS extension in TLS 1.2 ServerHello messages***].

4.2.3 Identification and Authentication (FIA)

4.2.3.1 Authentication Failure Management (FIA_AFL.1)

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [**1-100**] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

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

4.2.3.2 Password Management (FIA_PMG_EXT.1)

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [***"!", "@", "#", "$", "%", "^", "&", "*", "(", ")", ["'"], "'", "+", ",", "-", ".", "/", ":", ";", "<", "=", ">", "?", "[", "\", "]", "_", " " (i.e. empty space character), "`", "{", "|", "}", "~"]***];

b) Minimum password length shall be configurable to between [**8**] and [**32**] characters.

4.2.3.3 Protected Authentication Feedback (FIA_UAU.7)

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.

4.2.3.4 User Identification and Authentication (FIA_UIA_EXT.1)

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

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

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

FIA_UIA_EXT.1.3 The TSF shall provide the following remote authentication mechanisms [***Web GUI password, SSH password, SSH public key***] and local authentication mechanisms [***password-based***].

FIA_UIA_EXT.1.4 The TSF shall authenticate any administrative user's claimed identity according to each authentication mechanism specified in FIA_UIA_EXT.1.3.

4.2.3.5 X.509 Certificate Validation (FIA_X509_EXT.1/Rev)

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

- RFC 5280 certificate validation and certificate path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [***a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3***].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for DTLS/TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for DTLS/TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

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

4.2.3.6 X.509 Certificate Authentication (FIA_X509_EXT.2)

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

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

4.2.3.7 X.509 Certificate Requests (FIA_X509_EXT.3)

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

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

4.2.4 Security Management (FMT)

4.2.4.1 Management of Security Functions Behavior (FMT_MOF.1/ManualUpdate)

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

4.2.4.2 Management of TSF Data (FMT_MTD.1/CoreData)

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

4.2.4.3 Management of TSF Data (FMT_MTD.1/CryptoKeys)

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

4.2.4.4 Specification of Management Functions (FMT_SMF.1)

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

- Ability to administer the TOE remotely;
- Ability to configure the access banner;
- Ability to configure the remote session inactivity time before session termination;
- Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates;
- [
  - ***Ability to configure local audit behaviour (e.g. changes to storage locations for audit; changes to behaviour when local audit storage space is full, changes to local audit storage size);***
  - ***Ability to manage the cryptographic keys;***
  - ***Ability to configure the cryptographic functionality;***
  - ***Ability to set the time which is used for time-stamps;***
  - ***Ability to manage the TOE's trust store and designate X509.v3 certificates as trust anchors;***
  - ***Ability to generate Certificate Signing Request (CSR) and process CA certificate response;***
  - ***Ability to administer the TOE locally;***
  - ***Ability to configure the local session inactivity time before session termination or locking;***
  - ***Ability to configure the authentication failure parameters for FIA_AFL.1;***
  - ***Ability to manage the trusted public keys database;***

  ].

4.2.4.5 Restrictions on Security Roles (FMT_SMR.2)

FMT_SMR.2.1 The TSF shall maintain the roles:

- Security Administrator.

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

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

are satisfied.

4.2.5 Protection of the TSF (FPT)

4.2.5.1 Protection of Administrator Passwords (FPT_APW_EXT.1)

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

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

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

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

4.2.5.3 Reliable Time Stamps (FPT_STM_EXT.1)

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

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

4.2.5.4 TSF Testing (FPT_TST_EXT.1)

FPT_TST_EXT.1.1[4] The TSF shall run a suite of the following self-tests

- During initial start-up (on power on) to verify the integrity of the TOE firmware and software;
- Prior to providing any cryptographic service and [***on-demand, [periodically (when configured), Enabling FIPS-CC Mode]***] to verify correct operation of cryptographic implementation necessary to fulfil the TSF;
- [***start-up, on-demand, at the conditions [periodically (when configured), Enabling FIPS-CC Mode]***] self-tests [***Firmware Integrity, Configuration file integrity, Cryptographic Known Answer Tests***].

to demonstrate the correct operation of the TSF.

FPT_TST_EXT.1.2 The TSF shall respond to [***all failures***] by [***entering a maintenance mode, [and for self-test failures other than firmware integrity: when the TOE enters maintenance mode, the network interfaces are also brought down and the firmware OS is halted]***].

4.2.5.5 Trusted Update (FPT_TUD_EXT.1)

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

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

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

[4] Modified by TD0836

4.2.6 TOE Access (FTA)

4.2.6.1 TSF-Initiated Session Locking (FTA_SSL_EXT.1)

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

- ***terminate the session***]

after a Security Administrator-specified time period of inactivity.

4.2.6.2 TSF-Initiated Termination (FTA_SSL.3)

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

4.2.6.3 User-Initiated Termination (FTA_SSL.4)

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

4.2.6.4 Default TOE Access Banners (FTA_TAB.1)

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.

4.2.7 Trusted Path/Channels (FTP)

4.2.7.1 Inter-TSF Trusted Channel (FTP_ITC.1)

FTP_ITC.1.1 The TSF shall be capable of using [***TLS***] to provide a trusted communication channel between itself and authorized IT entities supporting the following capabilities: audit server, [***no other capabilities***] that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

FTP_ITC.1.2 The TSF shall permit [***the TSF***] to initiate communication via the trusted channel.

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

4.2.7.2 Trusted Path (FTP_TRP.1/Admin)

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

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

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

4.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are included by reference from the [cPPND].

Table 7: Assurance Components

Requirement Class	Requirement Component
ASE: Security Target	Conformance claims (ASE_CCL.1)
	Extended components definition (ASE_ECD.1)
	ST introduction (ASE_INT.1)
	Security objectives for the operational environment (ASE_OBJ.1)
	Stated security requirements (ASE_REQ.1)
	Security Problem Definition (ASE_SPD.1)
	TOE summary specification (ASE_TSS.1)
ADV: Development	Basic functional specification (ADV_FSP.1)
AGD: Guidance documents	Operational user guidance (AGD_OPE.1)
	Preparative procedures (AGD_PRE.1)
ALC: Life-cycle support	Labelling of the TOE (ALC_CMC.1)
	TOE CM coverage (ALC_CMS.1)
ATE: Tests	Independent testing – conformance (ATE_IND.1)
AVA: Vulnerability assessment	Vulnerability survey (AVA_VAN.1)

5 TOE Summary Specification

This section describes the following security functions implemented by the TOE to satisfy the SFRs claimed in Section 4.2:

- Security Audit
- Cryptographic Support
- Identification and Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channels.

5.1 Security Audit

5.1.1 Audit Data Generation

The TOE generates audit records for start-up and shut-down of the audit functions (in parallel with the device itself so device startup/shutdown is logged for this); and the administrative actions:

- Administrative login and logout.
- Changes to TSF data related to configuration changes (in addition to the information that a change occurred, the TOE logs what has been changed).
- Generating/import of, changing, or deleting of cryptographic keys (in addition to the action itself a unique key name or key reference is logged).
- Resetting passwords (name of related user account is logged).

Additionally, the TOE logs the specifically defined auditable events listed in Table 6.

The TOE records the following information within each audit record: date and time of the event, type of event, subject identity, the outcome (success or failure) of the event; and for each audit event type, based on the auditable event definitions of the functional components included in the cPP/ST, information specified in column three of Table 6.

All auditable events that involve configuration of the TSF include the action requested, the success or failure, and the identity of the user that made the request. This includes any cryptographic operations, specifically requesting to generate a certificate, generation and import of a CSR, import of a certificate, management of certificate stores, and managing the list of certificates used to validate servers. When auditing certificates, the TOE records reference identifier information such as hostname or IP address in order to uniquely identify the certificate. When auditing keys, the TOE records the name of the TOE user who performed the key operation and the location of the key file in order to uniquely identify the key.

This aspect of the Security Audit security function satisfies FAU_GEN.1 and FAU_GEN.2.

5.1.2 Audit Storage and Audit Record Export

The TOE is a single standalone appliance that stores audit logs locally and provides the administrator the ability to configure the real-time export of syslog records protected with TLS. To enable sending to a remote syslog server, use the command: *`config log syslogd setting`*. To enable local logging, use the *`log eventfilter`* command and the *`enable`* parameter for each category of event. The event type categories that include the required audit event records identified in the NDcPP are System and User.

Use the *config log memory global-setting* command to configure log threshold warnings, as well as the maximum buffer lines, for the FortiSwitch system memory.

Audit records are stored in the TOE's local system memory using syslog. Local log entries are not retained when the system restarts (the buffer is non-persistent), however the entries in the external syslog remain unchanged. The local memory buffer has a limited capacity and displays only the most recent log entries. After all available memory is used, the system begins to overwrite the oldest log messages.

The maximum size of the memory buffer is 98394 bytes by default but is configurable from 64KB to 64MB using the *set max-size* command. Three threshold warnings — first (default 75%), second (default 90%) and final (default 95%) — can be configured that will notify the administrator prior to the log disk becoming full. The thresholds can be configured as an integer within the following ranges: first- between 1 and 98; second- between 2 and 99; and third- between 3 and 100. When the memory buffer reaches the configured log threshold, an audit record will be written in the log buffer and also in the external syslog. There is a small software delay between when a log message is generated and is written to the syslog.

Audit records are protected against unauthorized modification and can only be viewed or deleted by a Security Administrator. The CLI command `log delete` clears all log entries in memory but does not affect the logs stored in the external syslog. There is no mechanism to modify logs and no other interfaces to access the audit records stored in memory, except for the log delete command.

This aspect of the Security Audit security function satisfies FAU_STG.1, FAU_STG_EXT.1, and FAU_STG_EXT.3.

5.2 Cryptographic Support

The TOE includes the FortiSwitch Crypto Library v7.6, which incorporates OpenSSL and OpenSSH. Table 8 below summarizes the CAVP certificates obtained for the TOE's cryptographic library. The TOE uses this library for verifying TOE update package signatures, for self-tests, and for all SSH, TLS and certificate functionality.

Table 8: Cryptographic Functions Implemented by OpenSSL/OpenSSH

Functions	Standards	Certificates
Asymmetric Key Generation (FCS_CKM.1)		
RSA (2048 bits, 3072 bits, 4096 bits)	FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.1	A6436
ECDSA (P-256, P-384, P-521 curves)	FIPS PUB 186-5, "Digital Signature Standard (DSS)", Appendix A.2	A6436
FFC 'safe-prime' (MODP Groups 14, 16, 18)	SP 800-56A Revision 3, RFC 3526	A6436
Key establishment (FCS_CKM.2)		
ECDSA Elliptic curve-based scheme (P-256, P-384, P-521)	NIST Special Publication 800-56A Revision 3	A6436
FFC 'safe-prime' (MODP Groups 14, 16, 18)	SP 800-56A Revision 3, RFC 3526	A6436
Data encryption (FCS_COP.1/DataEncryption)		

Functions	Standards	Certificates
AES in CBC mode (128, 256 bits) AES in GCM mode (128, 256 bits) AES in CTR mode (128, 256 bits)	ISO 18033-3 (AES) ISO 10116 (CBC mode) ISO 19772 (GCM mode) ISO 10116 (CTR mode)	A6436
Digital signature generation and verification (FCS_COP.1/SigGen)		
RSA Digital Signature Algorithm (2048 bit, 3072 bit, and 4096 bit modulus)	FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 5.4, using PKCS #1 v2.2 Signature Schemes RSASSA-PSS and/or RSASSAPKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	A6436
ECDSA Elliptic Curve Digital Signature Algorithm (P-256, P-384, P-521 curves)	FIPS PUB 186-5, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST Recommended" [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4	A6436
Cryptographic hashing (FCS_COP.1/Hash)		
SHA-1 (digest size 160 bits) SHA-256 (digest size 256 bits) SHA-384 (digest size 384 bits) SHA-512 (digest size 512 bits)	ISO/IEC 10118-3:2004	A6436
Keyed-hash message authentication (FCS_COP.1/KeyedHash)		
HMAC-SHA-1 (key size 160 bits, digest size 160 bits) HMAC-SHA-256 (key size 256 bits, digest size 256 bits) HMAC-SHA-384 (key size 384 bits, digest size 384 bits) HMAC-SHA-512 (key size 512 bits, digest size 512 bits)	ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"	A6436
Deterministic random bit generation (FCS_RBG_EXT.1)		
Counter DRBG (AES) -256 bits entropy	ISO/IEC 18031:2011	A6436

5.2.1 Cryptographic Operations

The TOE generates asymmetric keys for TLS, SSH and X.509 certificates as follows:

- TLS Server: 2048, 3072, and 4096 bits RSA, P-256, P-384, P-521 curves ECC
- TLS Client: 2048, 3072, and 4096 bits RSA, P-256, P-384, P-521 curves ECC
- X.509 Certificate Requests: 2048, 3072, and 4096 bits RSA, and P-256, P-384, P-521 curves ECC key generation
- SSH: P-256, P-384, P-521 ECC curves

The TOE performs key establishment for TLS and SSH as follows:

- TLS Server: ECDSA (P-256, P-384, P-521)
- TLS Client: ECDSA (P-256, P-384, P-521)
- SSH: ECDSA (P-256, P-384, P-521)

The TOE also implements the cryptographic algorithms listed below in support of TLS and SSH as well as for the additional functionality specified.

- AES-CBC, AES-GCM (128-bit, 256-bit): TLS only
  - AES-CBC-256 is also used for key encryption
- AES-CTR, AES-GCM (128-bit, 256-bit): SSH only
- RSA signature generation and verification (2048, 3072, and 4096 bits RSA): TLS only
  - 2048-bit RSA with SHA-256 is also used for TOE update, digital signature verification for firmware integrity, configuration file integrity, TOE update and self-tests
- ECDSA signature generation and verification (P-256, P-384, P-521): TLS and SSH
- SHA-1, SHA-256, SHA-384, SHA-512: TLS and SSH
  - SHA-256 is also used for administrator password hashing for non-volatile storage, digital signature verification for firmware integrity, configuration file integrity, TOE update and self-tests
  - SHA-1 is only used for TLS
  - SHA-512 is only used for SSH
- HMAC
  - HMAC-SHA-1 (160 bit key/512 block size, 160 bit output length): used for TLS,
  - HMAC-SHA-256 (256 bit key/512 block size, 256 bit output length), HMAC-SHA-384 (384 bit key/1024 block size, 384 bit output length): used for TLS
  - HMAC-SHA-256 (256 bit key/512 block size, 256 bit output length), HMAC-SHA-512 (512 bit key /1024 block size, 512 bit output length): used for SSH
  - HMAC-SHA-256 is also used for self-test of configuration integrity
- ecdsa-sha2-nistp521 is used for the SSH public-key based authentication.

This aspect of the Cryptographic Support security function satisfies FCS_COP.1/DataEncryption, FCS_COP.1/SigGen, FCS_COP.1/Hash, and FCS_COP.1/KeyedHash.

5.2.2 Random Bit Generation

The TOE uses a 256-bit AES-CTR DRBG, implemented in the FortiSwitch Crypto Library. The maximum security strength supported by the DRBG is the security strength of the block cipher (256 bits). The TOE instantiates the DRBG with full entropy (256 bits) obtained via JitterEntropy, which generates entropy from CPU execution time jitter. The collected entropy is injected into the Linux kernel `/dev/random` device using the `RNDADDENTROPY ioctl` for retrieval by the TOE's cryptographic module.

The TOE uses the DRBG to generate all keys (as part of SSH, TLS and CSR key generation) as well as to generate salts and nonces (for password hashing and TLS respectively).

This aspect of the Cryptographic Support security function satisfies FCS_RBG_EXT.1.

5.2.3 Cryptographic Key Generation and Establishment

The TOE supports generating key pairs, both for authentication and for key exchange for SSH and TLS. When generating authentication key pairs, the TOE can generate a CSR with RSA 2048, 3072, and 4096 bit key pairs; and ECDSA P-256, P-384, P521 key pairs. When generating ephemeral key pairs for key establishment, the TOE generates P-256/384/521 for both TLS and SSH; and DH MODP Groups 14, 16, and 18 for SSH.

For key establishment, the TOE will generate P-256/P-384/P-521 ECDHE keys (depending on the negotiated cipher suite) during TLS negotiation. The TOE acts as both a TLS server (to service incoming

administrative sessions) and as a TLS client (for syslog export). The TOE supports elliptic curve key establishment for TLS, depending on the negotiated cipher suite. The TOE's SSH implementation uses ecdh-sha2-nistp256, ecdh-sha2-nistp384, ecdh-sha2-nistp521, diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, and diffie-hellman-group18-sha512 for its key exchange methods.

This aspect of the Cryptographic Support security function satisfies FCS_CKM.1 and FCS_CKM.2.

5.2.4 Cryptographic Key Destruction

The TOE clears keys from volatile memory by overwriting the memory locations in RAM with zeroes. The TOE clears plaintext keys in non-volatile storage by performing a single overwrite consisting of either zeroes or a new value of the key, depending on the specific key (refer to the table below). Keys stored in flash are either stored in plaintext or AES-256-CBC encrypted as identified in the table below.

The encryption key for configuration data is generated from a password provided by the user and therefore is not persistently stored. When an administrator (optionally) creates an encrypted backup of the configuration, they must provide a password, which is used to create the key and encrypt the entire back up file. The same password must be entered to restore the configuration using the encrypted configuration file. The password used for backup is not stored on FSW. Its file encryption using the password, which means a salt/iv which is stored in the encrypted file is used, along with the password provided by the user for encryption providing the necessary key for the encryption operation on the backup file. For decryption since the salt/iv is in the file, the user provided password is used to generate the key for decryption.

Table 9: Key Clearing

Key	Storage Location	How Stored	Usage and destruction
SSH Host Private Key	Flash	plaintext	Used by SSH for client authentication of the SSH server, which is the TOE. It is replaced (overwritten) with a new key at request of the administrator using the command: `exec ssh-regen-keys`. The CLI accesses the underlying file system APIs.
SSH Session Key	RAM	plaintext	Used by SSH to encrypt a session and is destroyed when the SSH session is closed (single overwrite consisting of zeroes).
TLS pre-master secret	RAM	plaintext	Used by TLS and is destroyed when handshake is finished (single overwrite consisting of zeroes).
TLS session authentication key	RAM	plaintext	Used by TLS to authenticate a session and is destroyed when the TLS session

			is closed (single overwrite consisting of zeroes).
TLS Session encryption Key	RAM	plaintext	Used by TLS to encrypt a session and is destroyed when the TLS session is closed (single overwrite consisting of zeroes).
Diffie-Hellman Shared Secret	RAM	plaintext	Used by the SSH and TLS protocols and is destroyed when the session is closed (single overwrite consisting of zeroes).
Web Server Certificate Private Key	Flash & RAM	Encrypted, AES-256-CBC while in Flash (in the Backup Configuration file) 'live' copy is stored in plaintext in flash and loaded into RAM during use	Used for TLS web server and generated by user. When the administrator creates a new encrypted backup configuration file, the TOE uses file system APIs to destroy the private key in the backup configuration file by overwriting with a new key. The live private key in flash is destroyed when no longer needed (single overwrite consisting of zeroes) using the execute factoryreset command. The key in RAM is zeroized when no longer needed (single overwrite consisting of zeroes).

The TOE incorporates OpenSSL and OpenSSH cryptographic libraries which provide implementations of the cryptographic algorithms specified in Table 8. The TOE invokes the libraries' APIs to set up and maintain the full TLS/SSH session, using the underlying cryptographic algorithms as identified in Table 8. Therefore, all key generation, negotiation of session keys, and packet authentication is performed by the OpenSSL/OpenSSH cryptomodule. Files such as private keys and certificates are manually uploaded to the TOE during initial setup. Changes can be performed by remote access (CLI or GUI) or locally on the appliance with physical access to the serial ports.

The plaintext private keys and CSPs (see Table 9 above) are managed by the corresponding cryptomodules (e.g. keys associated with SSH are managed by OpenSSH, while keys associated with TLS are managed by OpenSSL) and stored in Flash and/or RAM/BIOS. Encrypted flash keys are copied unencrypted to a RAM drive at startup for runtime use. The cryptomodules do not store any other secret or private keys or CSPs persistently (beyond the lifetime of an API call). All secret keys, plaintext private keys and CSPs are destroyed by user command "factoryreset"; automatically by the APIs when no longer required by overwriting once with zeroes; or with a new value of the key (for SSH private keys).

All keys, key material, and authentication credentials are protected from unauthorized disclosure. There are no configurations or circumstances that do not conform to the key destruction requirement.

This aspect of the Cryptographic Support security function satisfies FCS_CKM.4.

5.2.5 Cryptographic Protocols

The TOE implements the following cryptographic protocols to protect communications between itself and non-TOE entities:

- SSH as a server—the TOE acts as an SSH server supporting inbound administrative sessions.
- TLS as a client—the TOE acts as a TLS client when exporting audit records to an external audit server.
- TLS as a server—the TOE acts as a TLS server supporting inbound administrative sessions.

Mutual authentication is not supported for TLS.

5.2.5.1 SSH Server Protocol

The TOE's SSH server implements the SSH protocol in accordance with RFCs 4251, 4252, 4253, 4254, 4256, 4344, 5647, 5656, 6668, and 8268. The TOE supports SSH password-based and public key-based authentication methods as described in RFC 4252, and keyboard-interactive authentication as described in RFC 4256. Keyboard-interactive is a generic authentication method that is used to implement different types of authentication methods in a manner that requires direct user input (e.g. typing a password). As described in RFC 4253, packets greater than 256 kilobytes in an SSH transport connection are dropped. Public-private keys must be created on the SSH client application. The public key is added to the appliance using the CLI.

For its SSH transport implementation the TOE uses the following encryption algorithms and rejects all other encryption algorithms: aes128-ctr, aes256-ctr, aes256-gcm@openssh.com and aes128-gcm@openssh.com.

The SSH public-key based authentication implementation uses ecdsa-sha2-nistp521 as its public key algorithm and rejects all other public key algorithms. When an SSH client presents a public key, the TOE verifies that the SSH client's presented public key matches one that is stored within the local account file.

The SSH transport implementation uses only hmac-sha2-256, hmac-sha2-512, and implicit GCM as its data integrity MAC algorithms and rejects all other MAC algorithms. Specifically, the TOE uses implicit GCM MACs for aes256-gcm@openssh.com and aes128-gcm@openssh.com.

The TOE's SSH implementation uses only diffie-hellman-group14-sha256, diffie-hellman-group16-sha512, and diffie-hellman-group18-sha512, ecdsa-sha2-nistp256, ecdsa-sha2-nistp384, and ecdsa-sha2-nistp521 for its key exchange methods and ecdsa-sha2-nistp521 to authenticate itself to its peer. The TOE's SSH KDF used to derive session keys from a shared secret is in accordance with RFC 5656 (Section 4).

The TOE manages a tracking mechanism for each SSH session so that it can initiate a new key exchange (rekey) when either 1 GB of data or 1 hour has passed, whichever threshold occurs first. The threshold limits apply to both transmitted and received data.

This aspect of the Cryptographic Support security function satisfies FCS_SSH_EXT.1 and FCS_SSHS_EXT.1.

5.2.5.2 TLS Client Protocol

The TOE's TLS client supports TLS 1.2 with the following TLS ciphersuites:

- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 8422
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 8422
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 8422
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 8422
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289

The TOE presents the signature_algorithms extension with support for the following algorithms only:

- ecdsa_secp256r1 with sha256(0x0403),
- ecdsa_secp384r1 with sha384(0x0503),
- ecdsa_secp521r1 with sha512(0x0603),
- rsa_pkcs1 with sha256(0x0401),
- rsa_pkcs1 with sha384(0x0501),
- rsa_pkcs1 with sha512(0x0601),
- rsa_pss_rsae with sha256(0x0804),
- rsa_pss_rsae with sha384(0x0805),
- rsa_pss_rsae with sha512(0x0806),
- rsa_pss_pss with sha256(0x0809),
- rsa_pss_pss with sha384(0x080a),
- rsa_pss_pss with sha512(0x080b)

The TOE does not support TLS 1.2 secure renegotiation attempts.

The supported ciphersuites are not configurable. The TOE prohibits the use of the early data extension and post-handshake client authentication according to RFC 8446, Section 4.2.6. TOE does not support or use pre-shared keys for its TLS implementation and denies versions of TLS older than TLS 1.2. This is done by default when placing the TOE into FIPS-CC mode.

The TOE allows the TLS client (used for audit export) to specify the syslog server by IP address (IPv4, IPv6) or fully qualified domain name. The TOE's TLS client implementation establishes its reference identifiers from the administrator-configured reference identifiers per Section 6 of RFC 6125, using the hostname or IPv4/IPv6 address in CN or IPv4/IPv6 address in the SAN as a reference identifier and checking that the syslog server's certificate includes the specified identifier. When the CN field contains a FQDN, wildcards are supported. When the CN field is composed of IPv4/IPv6 addresses, the TOE converts the text representation of the IP address in the CN to a binary representation of the IP address in network byte order in compliance with RFC 3986 or RFC 5952, respectively. The TLS client supports the Supported Groups extension (specifying only P-256, P-384, and P-521) in its Client Hello, and the TOE does not require (nor allow) administrative configuration of this extension; the TOE always sends it. When establishing a trusted channel, by default the TSF does not establish a trusted channel if the server certificate is invalid; there is no administrator override mechanism.

The TOE's TLS client does not support mutual authentication.

This aspect of the Cryptographic Support security function satisfies FCS_TLSC_EXT.1.

5.2.5.3 TLS Server Protocol

The TOE's TLS server supports TLS 1.2 with the same TLS ciphersuites as those identified for the TOE's TLS Client in Section 5.2.5.2. The supported ciphersuites are not configurable.

The TOE prohibits the use of the early data extension.

The TOE support secure renegotiation for TLS 1.2 in accordance with RFC 5746 by always including the "renegotiation_info" TLS extension in TLS 1.2 ServerHello messages.

The TOE does not support pre-shared keys for its TLS implementation and denies versions of TLS older than TLS 1.2. This is done by default when placing the TOE into FIPS-CC mode.

The TOE supports session resumption based on session tickets that are encrypted using 128, 256 AES-CBC/GCM as specified in FCS_COP.1/DataEncryption and that adhere to the structural format defined in section 4 of RFC 5077. Session tickets are enabled by default and the default session time out is set to 300 seconds. Reaching this threshold will trigger a full handshake for the TLS session. There is no global/inter-process SSL session cache. The TOE uses the session cache internal to OpenSSL that is for just one process, the httpsd parent process. This means that the id/ticket information can also be reused for different child processes.

The TOE uses an elliptic curve during TLS key exchange using P-256, P-384, or P-521. Mutual authentication is not performed and therefore certificate validation is not performed by the TOE's TLS server.

The TOE implements TLS server functionality for inbound management requests over HTTPS, therefore only the server requirements in RFC 2818 are applicable. The TOE conforms to RFC 2818 as follows: (section 2.1: complies as specified, section 2.2: complies as specified, section 2.2.1: not applicable, section 2.2.2: complies as specified, section 2.3: default port is 443, section 2.4: Use 'https' as specified, section 3.1: not applicable, and section 3.2: not applicable).

The TOE does not support mutual authentication for any of its HTTPS/TLS connections.

This aspect of the Cryptographic Support security function satisfies FCS_HTTPS_EXT.1 and FCS_TLSS_EXT.1.

5.3 Identification and Authentication

5.3.1 User Identification and Authentication

During initial configuration, the TOE's default admin password must be set and must conform to the configured password policy requirements. The default admin can define additional users. Username/password (for GUI and SSH), or public key credentials (for SSH) are necessary to log in and access the management functions. The local serial console connection only supports password based authentication.

During login, the password/key must match the ones defined in the local account. If the asserted identity, password, or key cannot be verified then the login fails and an audit record is generated.

The TOE does not allow any actions prior to requiring the identification and authentication process except the display and acceptance of the warning banner. The TOE provides no feedback to the administrator

during authentication other than the success or failure of their attempt. For local password-based credentials, the TOE accepts passwords that consist of any combination of uppercase letters, lowercase letters, numbers, and special characters. This includes all letters, numbers, symbols, punctuation, and space separators. The TOE supports characters: !"#$%&'()*+,-./:;<=>?@[\]^_`{|}~. The `config system password-policy` command is used to change the minimum password length to values between 8 and 32 characters (inclusive, where default is 8 characters).

This aspect of the Identification and Authentication security function satisfies FIA_UIA_EXT.1, FIA_UAU.7, and FIA_PMG_EXT.1.

5.3.2 Authentication Failure Management

For password-based authentication at both the CLI and GUI, the system enforces a default of three password retries, allowing the administrator a maximum of three attempts to log into their account before they are locked out for a set amount of time (60 seconds by default). The number of attempts can be configured to an alternate value between 1 and 100, as well as the wait time between 1 - 2147483647 seconds before the administrator can try to enter a password again.

If an authentication attempt exceeds the threshold, the TOE will enforce a lockout of the remote administrator for the specified time. The `admin-lockout-threshold` and `admin-lockout-duration` commands are used to configure the thresholds.

All administrative interfaces that use password-based authentication are subject to lockout. User authentication based on SSH public key does not enforce a lockout mechanism, so in the evaluated configuration it is necessary to ensure the configuration of at least one administrator account that uses public key authentication so that the TSF cannot be rendered inaccessible through excessive failed authentication attempts.

This aspect of the Identification and Authentication security function satisfies FIA_AFL.1.

5.3.3 X.509 Certificate Validation

The TOE performs X.509 certificate validation when a signed cert response to a CSR certificate is uploaded; for certificate-based user authentication; and for TLS server certificates presented to the TOE where the TOE acts as a TLS client. Revocation checking of certificates is also performed during TLS client connection establishment (i.e., when the TOE acts as a TLS client connecting to a remote syslog-tls server); using a certificate revocation list (CRL) as specified in RFC 5280 Section 6.3.

The TOE performs the revocation checking after having checked the validity of the server certificate and its chain, conformant to RFC 5280. This includes supporting a minimum path length of three certificates and the certification path terminating with a trusted CA certificate designated as a trust anchor. The TOE requires that TLS server certificates have the Server Authentication purpose and the TOE's own TLS client certificates have the Client Authentication purpose in order to be considered valid. The TOE will not treat a CA certificate as such unless the basicConstraints extension is present with the CA flag set to TRUE.

The TOE chooses the certificate to present to external TLS clients based on the server certificate that is loaded into it as part of administrative configuration. In all other cases the TOE validates the certificate that is presented to it and validates the chain based on what is included in the certificate.

The TOE does not use certificates for trusted updates or executable code integrity verification and therefore the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) is not checked in the extendedKeyUsage field. The TOE does not use OCSP to validate the revocation status of the certificate

and therefore the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field is not used.

This aspect of the Identification and Authentication security function satisfies FIA_X509_EXT.1/Rev.

5.3.4 X.509 Certificate Authentication

The TOE uses X.509 certificates for authentication of TLS and HTTPS trusted channels. The TOE relies upon the administrator to load the CA certificates (root CA and any needed intermediate certificates).

When receiving a connection from a remote administrator, the TSF will present its own server certificate to the client. Mutual authentication is not supported. During revocation checking, if the TOE cannot establish a connection to determine revocation status, the connection will be accepted.

This aspect of the Identification and Authentication security function satisfies FIA_X509_EXT.2.

5.3.5 X.509 Certificate Requests

The TOE allows an administrator to request the TOE perform on-board key generation of an RSA 2024/4096 key pair or ECDSA/scep256r1/384r1/521r1 and then outputs a Certificate Signing Request (CSR), which the administrator can have signed by a suitable CA. The CSRs are generated as specified by RFC 2986 and the administrator is able to provide the following information in the request: public key, Common Name, Organization, and Country.

This aspect of the Identification and Authentication security function satisfies FIA_X509_EXT.3.

5.4 Security Management

5.4.1 Security Roles and Specification of Management Functions

The TOE has one default “admin” account with all management capabilities including the ability to configure administrator accounts, adjust system settings, upgrade firmware, and configure security features. This account is a super_admin administrator profile that cannot be deleted or modified to ensure there is always a method to administer the FortiSwitch unit. This user profile has access to all components of the system, including the ability to add and remove other system administrators. For some administrative functions, such as backing up and restoring the configuration using SCP, super_admin access is required. There are no other default accounts or profiles. The admin can create other super_admin users or users with a subset of access permissions. This is accomplished by creating a ‘Profile’; assigning components to the profile; and then assigning the profile to a user. Administrator profiles can be configured using System > Admin > Profiles. Only one profile can be assigned to each user. Read/write permissions can be assigned to the following components: System Configuration, Network Configuration, Admin Users, Router Configuration, and Log & Report. Users can be assigned one or more of these permissions (but only one profile). Each permission is a subset of the super_admin role that has all permissions.

System Configuration provides access to the Dashboard, Network, Config, and certificate settings. These interfaces allow the admin to update the firmware, set the time, configure TLS, and configure certificates.

The Admin Users role has the ability to configure administrators, profiles, monitor and administrator settings such as the idle timeout. Users with this role also have the ability to disconnect (logout) other administrators

Log & Report access control allows access to view the event logs, log configuration (enable and disable), and configure the Syslog. A user with these permissions can also select which events to view by severity, user/process that generated the event, IP network service, action/event, or status.

The System Configuration, Admin Users, and Log & Report permissions provide a subset of the super_admin capabilities and therefore users with these permissions as well as the admin account correspond to the Security Administrator as defined in the PP.

The TOE provides the Security Administrator administrative access through its HTTPS server (GUI) and via a CLI. The same accounts/roles are used for both (i.e. a user account defined with a certain role can be used to access both the CLI and the GUI).

The CLI can be accessed remotely over SSH or locally over direct console connection to the TOE's dedicated management port (included on all TOE models). Administrator CLI authentication is performed using either username/password or username/public key, depending on the configuration of the administrator's account. The GUI can be accessed remotely over HTTPS or locally over direct console connection to the TOE's dedicated management port (i.e. both the CLI terminal server and the web server are accessible locally via this port). Administrator GUI authentication is performed using username/password.

The TOE can be administered locally and remotely. The TOE provides the following management functions:

- Configure the access banner
- Configure the remote session inactivity time before session termination
- Update the TOE and verify TOE updates prior to installation using digital signature verification
- Configure local audit behaviour (local audit storage size);
- Manage cryptographic keys: SSH host key and valid SSH public keys; CSRs; and TLS private key for web server
- Configure the cryptographic functionality
- Set the time used for time stamps
- Manage the TOE's trust store and designate X.509 certificates as trust anchors
- Generate Certificate Signing Request (CSR) and process CA certificate response;
- Configure the local session inactivity time before session termination;
- Configure the authentication failure parameters for FIA_AFL.1;
- Manage the trusted public keys database

All functionality is available at both the GUI and CLI except the following must be performed via the CLI: configuration of the access banner; authentication failure parameters; and manage the trusted public keys database.

This aspect of the Security Management security function satisfies FMT_SMR.2 and FMT_SMF.1.

5.4.2 Management of Security Functions Behavior

The ability to perform TOE updates is restricted to the Security Administrators with either System Configuration or super_admin permissions.

This aspect of the Security Management security function satisfies FMT_MOF.1/ManualUpdate.

5.4.3 Management of TSF Data

The TOE does not provide any access to management functions prior to authentication and restricts the ability to manage cryptographic keys to the Security Administrators with either System Configuration or super_admin permissions. Specifically, the Security Administrator may use the TOE to: load and manage the SSH public keys into the TOE's public key database; generate CSRs (which contain key pairs); and load (import) or generate certificates (whether it is a locally generated certificate for the TOE or one generated by an external CA; or a certificate used to validate a presented TLS client or server certificate) into the TOE's Trust Store. When generating a certificate, the Security Administrator can choose the key type (RSA or Elliptic Curve) and key size (2048 bits, 3072 bits, or 4096 bits for RSA; or SECP256R1, SECP384R1, or SECP521R1 curves for Elliptic curve). Certificate and SSH key management is available to authorized administrators from the CLI and the GUI.

This aspect of the Security Management security function satisfies FMT_MTD.1/CoreData and FMT_MTD.1/CryptoKeys.

5.5 Protection of the TSF

5.5.1 Protection of Administrator Passwords

The TOE stores administrative password data using a salted SHA-256 hash within an internal configuration file. The TOE does not provide interfaces for an administrator to view, extract, or read the password. The TOE only accepts passwords during authentication attempts (or during administrative changing of the password). The TOE hashes the provided password and then either compares it to the stored value or stores the new value depending upon whether the administrator is authenticating or changing the administrative password.

This aspect of the TSF Protection security function satisfies FPT_APW_EXT.1.

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

The TOE stores its plaintext private keys and CSPs (see Table 9 above) in Flash and/or RAM and does not provide any commands to access them. The private keys are used for the HTTPS and SSH management connections.

This aspect of the TSF Protection security function satisfies FPT_SKP_EXT.1.

5.5.3 TSF Testing

The TOE performs a series of power-up Known Answer Tests (a KAT for each library cryptographic algorithm that the TOE utilizes) as well as integrity testing during power-up, at the request of an administrator, and periodically. In particular, the TOE executes the following self tests.

Table 100: TSF Self-testing

KAT name	Description
AES	Advanced Encryption Standard (AES) self-test
RBG-instantiate	Random bit generator (RBG)-instantiate known answer test
RBG-reseed	RBG-reseed known answer test
RBG-generate	RBG-generate known answer test

RSA	Rivest, Shamir, and Adleman Algorithm (RSA) known answer test
SHA1-HMAC	SHA1-HMAC known answer tests
SHA256-HMAC	SHA256-HMAC known answer tests
SHA384-HMAC	SHA384-HMAC known answer tests
SHA512-HMAC	SHA512-HMAC known answer tests
ECDHE	ECDHE known answer test
Configuration	Configuration file integrity test
Firmware integrity test	Firmware integrity test

The self-test for verification of the integrity of the configuration file uses RSA 2048-bit signature and HMAC-SHA256 and for the TOE firmware uses RSA 2048-bit signature and SHA-256. For each self-test, the TOE uses known inputs to calculate an expected cryptographic result, and compares that result to the known result. If the calculated result matches the expected result, the test passes; if it does not match, the test fails. The results are displayed on screen and a log is generated for all self-tests. If any of the self-tests fail (startup, on-demand, conditional or when enabling FIPS-CC mode), the unit will enter a maintenance mode: *FIPS Error Mode*. For all self-test failures other than firmware integrity, when the unit enters *FIPS Error Mode,* any active network interfaces are brought down and the firmware OS is halted. The administrator must power cycle the unit to exit FIPS Error Mode.

The self-tests are run at power-up, when FIPS-CC mode is enabled, at the request of the administrator, and can be configured to run periodically during normal operation. FIPS-CC mode is enabled during initial configuration from the local console using the default 'admin' account. This user account initially does not have a password. When FIPS-CC mode is enabled, the TOE will prompt for a password and reboot. The TOE reboots, operating in FIPS-CC mode and executes the start-up self-tests.

The commands available for executing the on-demand self-tests are **`execute fips kat all`** (to initiate all self-tests) and **`execute fips kat <test>`** (to initiate a specific self-test). To configure a self-test to periodically run, use **`config system fips-cc;`** from here enter values for **`reseed-interval`** and **`self-test-interval`** and set the function to "**`enable`**".

The self-tests are sufficient to demonstrate that the TSF is operating correctly since they encompass the cryptographic functionality and integrity testing of the entire TOE firmware executable code as well as the configuration file.

This aspect of the TSF Protection security function satisfies FPT_TST_EXT.1.

5.5.4 Trusted Update

The TOE provides an administrator the ability to view the currently executing version of firmware, as well as any firmware that is installed but inactive, on the System Configuration page within the GUI (*`System > Config > Firmware`*). The CLI can also be used to identify the currently executing firmware via the `get system status` command and to identify all installed firmware images via the `diagnose sys flash` command.

The administrator can manually initiate updates to the TOE firmware from these same pages by selecting Upgrade from the dashboard or selecting *`Choose File`* from the System Configuration page and then

navigating to the firmware image (and then *apply*). Updates can also be initiated using the CLI command: *restore image*.

The images are obtained from the Fortinet Support website:
https://support.fortinet.com/Download/FirmwareImages.aspx.

Once a firmware upgrade request has been initiated and prior to installation, the TOE will automatically verify the integrity of the image using RSA 2048-bit digital signature and SHA256, regardless of whether the update was initiated from the GUI or from the CLI. If the signatures verify, the TOE downloads the firmware image, reboots and then loads the new firmware, otherwise the upgrade is rejected. TOE functions other than the upgrade process will not be available during this process (reboot and firmware loading). Once the firmware is loaded, the TOE is operational.

From the web interface, it is not possible to download an update and then install it at a later date. An image is stored in the current running partition which can be primary or backup partition. The file is not active until the administrator reboots the TOE. Once rebooted, the TOE installs it and it is immediately activated.

The TOE allows two different images to be booted from primary and secondary flash partitions. Using the command: *set-next-reboot*, a TOE update could potentially be loaded into primary/secondary flash partitions that would then be installed at the next reboot.

In addition to the automatic image verification performed by the TOE, the admin can also verify image integrity using the command *verify image primary | secondary*

When an upgrade fails the device remains in the current version and this can be verified by running the *get system status* command. All failures are recorded in the syslog.

This aspect of the TSF Protection security function satisfies FPT_TUD_EXT.1.

5.5.5 Reliable Time Stamps

The TOE utilizes time when creating audit records and when checking syslog server and administrative client certificates (for expiration and revocation), for session inactivity timeout, for SSH time-based rekeying, and for administrator lock out periods. The TOE obtains and maintains time by allowing the administrator to manually specify the time. The TOE models each have a hardware real-time clock that is used to guarantee the availability of time data.

This aspect of the TSF Protection security function satisfies FPT_STM_EXT.1.

5.6 TOE Access

5.6.1 Access Banner

The TOE provides the administrator the ability to set a banner that the TOE displays before each local and remote administrator login. Remote administrator logins occur through the TOE's SSH or TLS-protected management ports or locally via direct connection. The TOE provides a pre-login banner that when configured is displayed before the user enters their user name and password. The same configured banner is used for both CLI and GUI. The banners are configured using the CLI: *config system global: set pre-login-banner* command.

This aspect of the TOE Access security function satisfies FTA_TAB.1.

5.6.2 Session Termination

The TOE terminates local and remote inactive sessions when the administrative configurable inactive timeout value is reached. This is configured from either the GUI: *System > Admin > Settings* or from the CLI: *admintimeout <admin_timeout_minutes>*. The default is 5 minutes and can be configured to values between 1 and 480 minutes. The settings are applied to both local and remote connections for both CLI and GUI. The configured timeout will apply to all ports specified and takes effect immediately.

The Administrator can disconnect their administrative session from the console using the 'exit' command. From the web GUI admin menu in the upper right page banner, the administrator can end their session by selecting *logout*.

This aspect of the TOE Access security function satisfies FTA_SSL_EXT.1, FTA_SSL.3, and FTA_SSL.4.

5.7 Trusted Path/Channels

The TOE provides the administrator the ability to export audit records to a TLS-protected syslog server. The TOE's TLS syslog configuration allows the administrator to specify the root CA certificate (as well as any needed intermediate CA certificates and any CRLs for revocation) for the TOE to use when validating the syslog server's certificate.

The administrator can connect to the TOE using its TLS/HTTPS or SSH protected administration channel.

The Trusted Path/Channels security function satisfies FTP_ITC.1 and FTP_TRP.1/Admin.

6 Protection Profile Claims

This ST conforms to the collaborative Protection Profile for Network Devices, Version 3.0e, 06 December 2023 [cPPND] and including the following optional and selection-based SFRs: FAU_STG.1, FAU_STG_EXT.3; FCS_HTTPS_EXT.1; FCS_TLSC_EXT.1; FCS_TLSS_EXT.1; FIA_AFL.1; FIA_PMG_EXT.1; FIA_UAU.7; FIA_X509_EXT.1/Rev; FIA_X509_EXT.2; FIA_X509_EXT.3; FMT_MTD.1/CryptoKeys; FPT_APW_EXT.1; and FTA_SSL_EXT.1; and Functional Package for Secure Shell (SSH), Version 1.0, 13 May 2021 [PKG_SSH], including the following selection-based SFRs: FCS_SSHS_EXT.1.

As explained in Section 2, Security Problem Definition, the Security Problem Definition of the [cPPND] has been included by reference into this ST.

As explained in Section 3, Security Objectives, the ST reproduces the security objectives for the operational environment from [cPPND].

As explained in Section 4, IT Security Requirements, the SFRs have all been drawn from [cPPND] and [PKG_SSH]. As such, operations on SFRs already performed in that PP are not identified in this ST. Rather, the SFRs have been copied from [cPPND] and [PKG_SSH] and any formatting used in the PP/Package has been removed. Operations performed on SFRs in the writing of this ST are identified in accordance with the conventions described in Section 1.3.

The SARs for the TOE are included by reference from the [cPPND].

7 Rationale

This ST includes by reference the [cPPND] Security Problem Definition and SARs and reproduces the security objectives for the Operational Environment. The ST makes no additions to the [cPPND] assumptions. [cPPND] and [PKG_SSH] SFRs have been reproduced with the Protection Profile/Package operations completed. Operations on the SFRs follow [cPPND] or [PKG_SSH] application notes and assurance activities. Consequently, [cPPND] rationale applies but is incomplete. The TOE Summary Specification rationale below serves to complete the rationale required for the security target.

7.1 TOE Summary Specification Rationale

Each subsection in Section 5, the TOE Summary Specification (TSS), describes a security function of the TOE. Each description identifies the SFRs that are covered by that description and, as such, provides the rationale that indicates which requirements are satisfied by aspects of the corresponding security function. The security functions work together to satisfy all of the security functional requirements. Furthermore, all of the security functions are necessary in order for the TSF to provide the required security functionality.

This section, in conjunction with the TSS, provides evidence that the security functions are suitable to meet the TOE security requirements. The collection of security functions work together to provide all of the security requirements. The security functions described in the TSS are all necessary for the required security functionality in the TSF. Table 111: Security Functions vs. Requirements Mapping summarizes the relationship between security requirements and security functions.

Table 111: Security Functions vs. Requirements Mapping

	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE access	Trusted path/channels
FAU_GEN.1	X						
FAU_GEN.2	X						
FAU_STG.1	X						
FAU_STG_EXT.1	X						
FAU_STG_EXT.3	X						
FCS_CKM.1		X					
FCS_CKM.2		X					
FCS_CKM.4		X					
FCS_COP.1/DataEncryption		X					
FCS_COP.1/SigGen		X					
FCS_COP.1/Hash		X					
FCS_COP.1/KeyedHash		X					

	Security audit	Cryptographic support	Identification and authentication	Security management	Protection of the TSF	TOE access	Trusted path/channels
FCS_HTTPS_EXT.1		X					
FCS_RBG_EXT.1		X					
FCS_SSH_EXT.1		X					
FCS_SSHS_EXT.1		X					
FCS_TLSC_EXT.1		X					
FCS_TLSS_EXT.1		X					
FIA_AFL_EXT.1			X				
FIA_PMG_EXT.1			X				
FIA_UIA_EXT.1			X				
FIA_UAU.7			X				
FIA_X509_EXT.1/Rev			X				
FIA_X509_EXT.2			X				
FIA_X509_EXT.3			X				
FMT_MOF.1/ManualUpdate				X			
FMT_MTD.1/CoreData				X			
FMT_MTD.1/CryptoKeys				X			
FMT_SMF.1				X			
FMT_SMR.2				X			
FPT_APW_EXT.1					X		
FPT_SKP_EXT.1					X		
FPT_TST_EXT.1					X		
FPT_TUD_EXT.1					X		
FPT_STM_EXT.1					X		
FTA_SSL_EXT.1						X	
FTA_SSL.3						X	
FTA_SSL.4						X	
FTA_TAB.1						X	
FTP_ITC.1							X
FTP_TRP.1/Admin							X