Fortinet FortiMail Version 7.4

Security Target

Version 1.0

19 April 2024

Prepared for:

Fortinet, Inc.
899 Kifer Road
Sunnyvale, CA 94086 USA

Prepared by:

Accredited Testing and Evaluation Labs
6841 Benjamin Franklin Drive
Columbia, MD 21046

Contents

Tables

1 Security Target Introduction

The Security Target (ST) contains the following additional sections:

- Product and TOE Description (Section 2)
- Excluded Functionality
- The TSF is being evaluated as a network device and therefore the email security system providing multi-layered protection against blended threats comprised of spam, viruses, worms and spyware capabilities was not evaluated. 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 2: Excluded Functionality

Feature	Description
Email security system (antivirus, antispam, content filtering, email routing and email archiving)	FotiMail provides an email security system, multi-layered protection against blended threats comprised of spam, viruses, worms and spyware functions. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Inbound filtering engine	FortiMail's inbound filtering engine blocks spam and malware before it can clog networks and affect users. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Dynamic and static user-blocking	FortiMail's dynamic and static user-blocking provides granular control over all email policies and users. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Predefined or customized dictionaries	FortiMail's predefined or customized dictionaries prevent accidental and intentional loss of confidential data. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Identity-Based Encryption (IBE), S/MIME, TLS email encryption	Secure content delivery is enforced with FortiMail's Identity-Based Encryption (IBE), S/MIME, or TLS email encryption options. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Mail Transfer Agent (MTA)	The [NDcPP] does not define requirements for MTA and therefore it has not been evaluated.

Feature	Description
Non FIPS-CC mode of operation	The TOE runs in FIPS-CC mode of operation. Non FIPS-CC mode of operation is excluded from the evaluation configuration.
Any features not associated with SFRs in claimed [cPPND]	[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.

- Security Problem Definition (Section 2.6)
- Security Objectives (Section 4)
- IT Security Requirements (Section 5)
- TOE Summary Specification (Section 6)
- Protection Profile Claims (Section 7)
- Rationale (Section 8)

1.1 Security Target, TOE, and CC Identification

ST Title – Fortinet FortiMail Version 7.4 Security Target

ST Version – Version 1.0

ST Date – 19 April 2024

TOE Identification – Fortinet FortiMail Version 7.4

The TOE consists of the following appliances running FortiMail firmware v7.4:

- FortiMail 200F (FML-200F)
- FortiMail 400F (FML-400F)
- FortiMail 900F (FML-900F)
- FortiMail (FML-2000F)
- FortiMail (FML-3000F)
- FortiMail VM Virtual Machine (VM)

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:

- *collaborative Protection Profile for Network Devices, Version 2.2e, 23 March 2020* (NDcPP) with the following optional and selection-based SFRs:
  - FCS_HTTPS_EXT.1
  - FCS_SSHS_EXT.1
  - FCS_TLSC_EXT.1
  - FCS_TLSC_EXT.2

- FCS_TLSS_EXT.1
- FIA_X509_EXT.1/Rev
- FIA_X509_EXT.2
- FIA_X509_EXT.3
- FMT_MOF.1/Functions
- FMT_MTD.1/CryptoKeys

- The following NIAP Technical Decisions apply to the TOE and have been accounted for in the ST development and the conduct of the evaluation, or were considered to be non-applicable:

  TD0527: Updates to Certificate Revocation Testing (FIA_X509_EXT.1)

  - This TD is applicable to the TOE but applies specifically to test activities so the ST itself is unaffected.

  TD0528: NIT Technical Decision for Missing EAs for FCS_NTP_EXT.1.4

  - This TD is not applicable to the TOE; the TOE does not claim FCS_NTP_EXT.1.

  TD0536: NIT Technical Decision for Update Verification Inconsistency

  - This TD is applicable to the TOE but applies specifically to test activities so the ST itself is unaffected.

  TD0537: NIT Technical Decision for Incorrect Reference to FCS_TLSC_EXT.2.3

  - This TD is applicable to the TOE. It corrects a reference to non-existent FCS_TLSC_EXT.2.3 in the PP, but does not affect any content in the ST.

  TD0546: NIT Technical Decision for DTLS – Clarification of Application Note 63

  - This TD is not applicable to the TOE. The TD applies to DTLS requirements, which the TOE does not claim.

  TD0547: NIT Technical Decision for Clarification on Developer Disclosure of AVA_VAN

  - This TD is applicable to the TOE but applies specifically to test activities so the ST itself is unaffected.

  TD0555: NIT Technical Decision for RFC Reference Incorrect in TLSS Test

  - This TD is applicable to the TOE but applies specifically to test activities so the ST itself is unaffected.

  TD0556: NIT Technical Decision for RFC 5077 Question

  - This TD is applicable to the TOE.

  TD0563: NIT Technical Decision for Clarification of Audit Date Information

  - This TD is applicable to the TOE but applies specifically to an application note in the PP for an SFR that the TOE claims so the ST itself is unaffected.

  TD0564: NIT Technical Decision for Vulnerability Analysis Search Criteria

- This TD is applicable to the TOE but applies specifically to test activities so the ST itself is unaffected.

TD0569: NIT Technical Decision for Session ID Usage Conflict in FCS_DTLSS_EXT.1.7

- This TD is applicable to the TOE.

TD0570: NIT Technical Decision for Clarification About FIA_AFL.1

- This TD is applicable to the TOE.

TD0571: NIT Technical Decision for Guidance on How to Handle FIA_AFL.1

- This TD is applicable to the TOE, but it only affects how a claimed SFR is interpreted so the ST itself is unaffected.

TD0572: NIT Technical Decision for Restricting FTP_ITC.1 to Only IP Address Identifiers

- This TD is applicable to the TOE, but it only affects how a claimed SFR is interpreted so the ST itself is unaffected.

TD0580: NIT Technical Decision for clarification about use of DH14 in NDcPPv2.2e

- This TD is applicable to the TOE.

TD0581: NIT Technical Decision for Elliptic curve-based key establishment and NIST SP 800-56Arev3

- This TD is applicable to the TOE.

TD0591: NIT Technical Decision for Virtual TOEs and Hypervisors

- This TD is applicable to the TOE but there is no change that directly affects the content of the ST.

TD0592: NIT Technical Decision for Local Storage of Audit Records

- This TD is applicable to the TOE but there is no change that directly affects the content of the ST.

TD0631: NIT Technical Decision for Clarification of public key authentication for SSH Server

- This TD modifies FCS_SSHS_EXT.1 and FMT_SMF.1, and is applicable to the TOE.

TD0632: NIT Technical Decision for Consistency with Time Data for vNDs

- This TD is not applicable to the TOE since it does not use the virtualization system as an external time source for synchronization.

TD0635: NIT Technical Decision for TLS Server and Key Agreement Parameters

- This TD is applicable to the TOE but there is no change that directly affects the contents of the ST.

TD0636: NIT Technical Decision for Public Key User Authentication for SSH

- The ST does not claim FCS_SSHC_EXT.1 and therefore this TD is not applicable.

TD0638: NIT Technical Decision for Key Pair Generation for Authentication

- This TD is applicable to the TOE.

TD0639: NIT Technical Decision for NTP MAC Keys

- This TD is not applicable to the TOE. The TOE does not claim NTP functionality.

TD0670: NIT Technical Decision for Mutual and Non-Mutual TLSC testing

- This TD is applicable to the TOE.

TD0738: NIT Technical Decision for Link to Allowed-With List

- This TD is an administrative action that moves the allowed-with lists to GitHub. It is not applicable to the TOE or to the ST.

TD0790: NIT Technical Decision: Clarification Required for testing IPv6

- This TD is applicable to the TOE, however it only modifies a test and therefore is not applicable to the ST.

TD0792: NIT Technical Decision: FIA_PMG_EXT.1 - TSS EA not in line with SFR

- This TD is applicable to the TOE however it only modifies an evaluation activity and therefore does not affect the ST.

TD0800: Updated NIT Technical Decision for IPsec IKE/SA Lifetimes Tolerance

- This TD is not applicable to the TOE since it does not use IPsec.

- 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

1.3 Conventions

The following conventions have been applied in this document:

- Security Functional Requirements – Part 2 of the CC defines the approved set of operations that may be applied to functional requirements: iteration, assignment, selection, and refinement.
  - Iteration: allows a component to be used more than once with varying operations. An iterated SFR is indicated by adding a string starting with '/' (e.g. 'FCS_COP.1/Hash').
  - Assignment: allows the specification of an identified parameter. Assignments are indicated using italics and are surrounded by brackets (e.g., [*assignment item*]). Note that an assignment within a selection would be identified in both italics and underline, with the brackets themselves underlined since they are explicitly part of the selection text, unlike the brackets around the selection itself (e.g., [selection item, [*assignment item inside selection*]]).

  - Selection: allows the specification of one or more elements from a list. Selections are indicated using underlines and are surrounded by brackets (e.g., [selection item]).
  - Refinement: allows technical changes to a requirement to make it more restrictive and allows non-technical changes to grammar and formatting. Refinements are indicated using bold, for additions, and strike-through, for deletions (e.g., "… **all** objects …" or "… ~~some~~ **big** things …"). Note that minor grammatical changes that do not involve the addition or removal of entire words (e.g., for consistency of quantity such as changing "meets" to "meet") do not have formatting applied.

- Other sections of the ST – Other sections of the ST use bolding to highlight text of special interest, such as captions.
- The ST does not show selection/assignment operations that have been completed by the PP authors, though it does preserve brackets to show where such operations have been made.

1.3.1 Terminology

This ST does not use any product specific terms.

1.3.2 Acronyms

Table 1: Acronyms

Term	Definition
DRBG	Deterministic Random Bit Generator
DSS	Digital Signature Standard
ECC	Elliptic Curve Cryptography
ECDH	Elliptic Curve Diffie Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
FAZ	Fortinet FortiAnalyzer
FQDN	Fully Qualified Domain Name
GCM	Galois Counter Mode
IBE	Identity-Based Encryption
MTA	Mail Transfer Agent
NTP	Network Time Protocol
PP	Protection Profile
RBG	Random Bit Generator
rDSA	RSA Digital Signature Algorithm
RNG	Random Number Generation
S/MIME	Secure/Multipurpose internet Mail Extensions
SMTP(S)	Simple Mail Transfer Protocol (Secure)
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality

2 Product and TOE Description

2.1 Introduction

Fortinet FortiMail is a specialized email security system that provides multi-layered protection against blended threats comprised of spam, viruses, worms and spyware. FortiMail's inbound filtering engine blocks spam and malware before it can clog networks and affect users. FortiMail's dynamic and static user-blocking provides granular control over all email policies and users. Secure content delivery is enforced with FortiMail's Identity-Based Encryption (IBE), S/MIME, or TLS email encryption options. FortiMail's predefined or customized dictionaries prevent accidental and intentional loss of confidential data.

For this Security Target, the Target of Evaluation (TOE) is FortiMail evaluated as a network device. The TOE claims exact conformance to the NDcPP. As such, the security-relevant functionality of the product is limited to the claimed requirements in this PP. The security-relevant functionality is described in sections 2.3 and 2.4. The product overview in section 2.2 below is intended to provide the reader with an overall summary of the entire product so that its intended usage is clear. The subset of the product functionality that is within the evaluation scope is subsequently described in the sections that follow it.

2.2 Product Overview

FortiMail supports three modes of operation: gateway mode, transparent mode and server mode. Gateway mode and transparent mode are within the scope of this evaluation. In all modes, the FortiMail system provides antivirus, antispam, content filtering, email routing and email archiving functionality with only minor changes to existing networks. These features are not within the scope of this evaluation.

When operating in gateway mode, FortiMail acts as a Mail Transfer Agent (MTA), also known as an email gateway or relay. When operating in gateway mode, all of the system's interfaces are on different IP subnets and the FortiMail acts as a router for SMTP/SMTPS traffic. When operating in transparent mode, FortiMail acts as either an implicit relay or a proxy. By default, FortiMail units operating in transparent mode are configured as a bridge, with all network interfaces on the same subnet. The FortiMail system receives email messages, scans for viruses and spam, then relays email to its destination email server for delivery. External MTAs connect to the FortiMail system, rather than directly to the protected email server. MTA was not covered within the scope of this evaluation. Note, these modes relate to the TOEs position in the deployed network and not to the evaluated functionality.

2.3 TOE Overview

The TOE is Fortinet FortiMail Version 7.4. Specifically, the TOE consists of hardware and virtual network device models (identified in Section 1.1 and described in Section 2.4) and FortiMail OS version 7.4 (i.e. the firmware) that runs on it. The virtualized TOE is a "Type 1" virtual network device as defined by section 1.2 of the NDcPP. The TOE boundary of the virtual model therefore includes only the virtualized network device, while its underlying hypervisor and physical platform are environmental.

With respect to the security functionality of the TOE, the TSF is limited to the relevant functionality that is defined in the claimed PP. The logical boundary of the TOE is summarized in section 2.4.2. However, the following general capabilities are considered to be within the scope of the TOE:

- **Protection of data in transit:** the TOE secures data in transit between itself and its operational environment using TLS and HTTPS.

- **Cryptographic services:** the TOE includes libraries with NIST-validated algorithm services that it uses for secure protocol implementations. It also provides X.509 certificate services in support of these protocols.
- **Identification, authentication, administration, and accounting:** the TOE includes mechanisms for authenticating remote and local administrators to facilitate authorized access to its management functions and recording the security-relevant actions that occur.
- **Self-protection:** the TOE includes various self-protection mechanisms to reduce the risk that the TSF or its data have their functionality altered through deliberate or accidental means.

2.4 TOE Architecture

The TOE consists of the following hardware and virtual network devices, each running FortiMail firmware v7.4:

Model	CPU	Storage	RAM
FML-200F	Intel Celeron G3900 Skylake, 2.80 GHZ	2x 2TB HDD	64GB
FML-400F	Intel i3-6100 Skylake, 3.7 GHz	2x 4TB HDD	64GB
FML-900F	Intel E3-1275V6 Kaby Lake, 3.80GHz	4X 4TB HDD	64GB
FML-2000F	Intel Xeon Silver 4210R Cascade Lake, 2.4GHz	10X 20TB	1TB
FML-3000F	Intel Xeon Silver 4210R Cascade Lake, 2.4GHz (dual CPU)	10X 20TB	1TB
VM	1 vCPU	50GB minimum 1TB Maximum	1GB

FortiMail includes Linux kernel 5.10.180 with OpenSSL 1.1.1w.

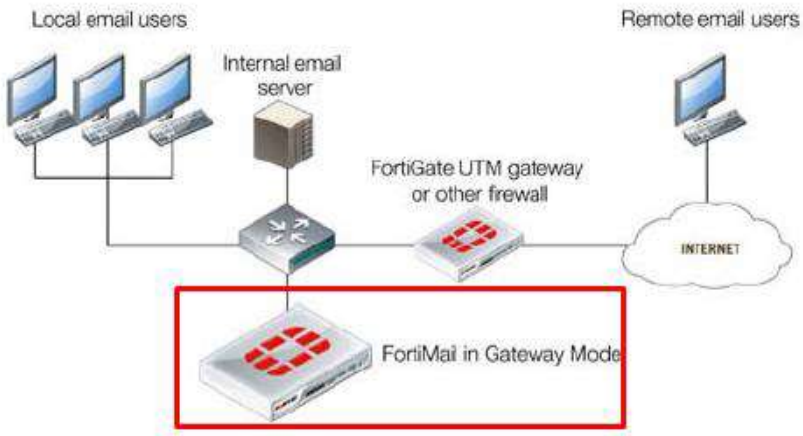
The FortiMail virtual network device runs on an environmental hypervisor (VMware ESXi v8.0) and was tested on an Intel Xeon E5-2620v4, 8 Cores, 2.10GHz.

2.4.1 Physical Boundary

The TOE consists of one of the Fortinet FortiMail appliances identified above as well as its firmware. The firmware version 7.4 is pre-installed on the TOE hardware. The firmware images for the virtualized appliance are downloadable from the Fortinet website. The network on which the TOE resides is considered part of the environment.

Gateway mode and transparent mode are within the scope of this evaluation. Figure 1 shows the TOE in a sample gateway mode deployment with its operational environment. TOE boundary is surrounded with red lines.

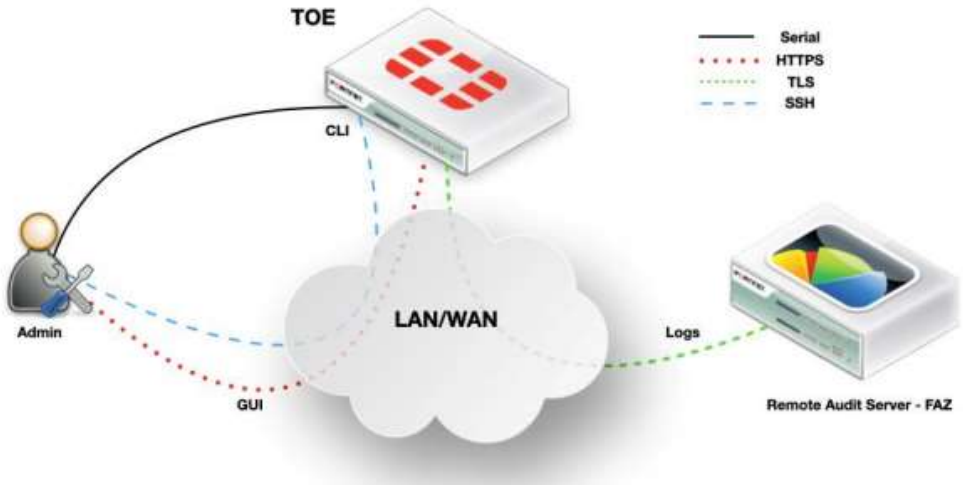
Figure 1 - TOE Boundary (when deployed in gateway mode)



As previously stated in section 2.2, transparent mode differs from gateway mode by its placement on a single subnet rather than routing between networks. Note that FortiMail is not a general purpose router. It can act as a mail gateway between networks, but does not route other traffic.

Figure 2 shows the TOE management interfaces as well as the interface with the remote audit server.

Figure 2 - TOE Interfaces



The TOE interfaces are as follows:

- CLI: Administrative CLI via direct serial connection or SSH.
- GUI: Administrative web GUI via HTTPS.
- Logs: Forwarding of TOE audit events to a remote audit server, which is a Fortinet FortiAnalyzer (FAZ), via TLS.

The TOE's operational environment includes the following:

- Remote audit server
- Platform (hardware and firmware) on which the virtual appliance TOE is hosted. In the tested configuration, this included the following:

  - VMware ESXi 8.0
  - Intel Xeon E5-2620V4, 8 Cores, 2.10GHz processor (Broadwell) processor
- Access to a Certification Authority and corresponding revocation checking mechanism for certificate management.
- A remote management workstation with a supported web browser for remote administrative access:
  - The tested configuration used Google Chrome 117.0.5938.150

The TOE runs in FIPS-CC mode of operation. Non FIPS-CC mode of operation is excluded from the evaluation configuration.

2.4.2 Logical Boundary

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

2.4.2.1 Security Audit

The TOE generates audit records of security-relevant activity. Audit data is stored locally and the TOE also has the ability to export all audit records to an external audit server over a TLS protected channel. The TOE manages the audit storage by overwriting previous audit records when the local storage space for audit data is full in order to capture new events.

2.4.2.2 Cryptographic Support

The TOE implements cryptographic functions in support of trusted communications, key pair generation for X.509 certificate requests, and self-testing. For trusted communications, the TOE implements TLS as a server with HTTPS, and TLS as a client without HTTPS. The TOE's TLS client supports mutual authentication. The TOE relies on platform hardware to generate entropy that is used to seed its DRBG to ensure that generated keys have the advertised security strength.

2.4.2.3 Identification and Authentication

The TOE uses a local password-based mechanism and additionally for SSH, an SSH public key-based mechanism for administrator authentication. The TOE enforces restrictions on the length and character composition of administrator passwords. Excessive failed authentication attempts on a remote administrative interface will cause a lockout that is resolved by a waiting period. The TOE also uses X.509 certificates for authentication of TLS connections. The TOE has a mechanism by which a certificate signing request can be generated so that it may obtain a certificate for its own use from a trusted CA.

2.4.2.4 Security Management

The TOE has a web-based remote management interface as well as a local/remote console. Most functionality can be administered over both interfaces. The TOE uses a single Security Administrator role to authorize the use of management functions.

2.4.2.5 Protection of the TSF

The TOE protects sensitive data from unauthorized access. It enforces integrity of its own contents through the use of self-tests to ensure that the TSF has not been modified. Firmware updates are obtained through the operational environment (e.g. downloaded from the vendor's support site); updates have a digital signature that is verified prior to application.

2.4.2.6 TOE Access

The TOE controls access through enforcement of idle session timeout on its management interfaces. These interfaces also display a configurable pre-authentication warning banner that advises against unauthorized use of the TOE.

2.4.2.7 Trusted Path/Channels

The TOE implements a TLS trusted channel between itself and trusted external audit servers. The TOE also implements SSH and TLS/HTTPS trusted paths for secure remote administration.

2.5 TOE Documentation

Fortinet provides the following product documentation in support of the installation and secure use of the TOE:

- Fortinet FortiMail 7.4 FIPS 140-3 and Common Criteria Technote, Version 0.2, April 19, 2024.
- Fortinet FortiMail 7.4.1 CLI Reference, February 7, 2024.
- Fortinet FortiMail 7.4.1 Administration Guide, April 12, 2024.

2.6 Excluded Functionality

The TSF is being evaluated as a network device and therefore the email security system providing multi-layered protection against blended threats comprised of spam, viruses, worms and spyware capabilities was not evaluated. 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 2: Excluded Functionality

Feature	Description
Email security system (antivirus, antispam, content filtering, email routing and email archiving)	FotiMail provides an email security system, multi-layered protection against blended threats comprised of spam, viruses, worms and spyware functions. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Inbound filtering engine	FortiMail's inbound filtering engine blocks spam and malware before it can clog networks and affect users. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Dynamic and static user-blocking	FortiMail's dynamic and static user-blocking provides granular control over all email policies and users. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Predefined or customized dictionaries	FortiMail's predefined or customized dictionaries prevent accidental and intentional loss of confidential data. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Identity-Based Encryption (IBE), S/MIME, TLS email encryption	Secure content delivery is enforced with FortiMail's Identity-Based Encryption (IBE), S/MIME, or TLS email encryption options. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Mail Transfer Agent (MTA)	The [NDcPP] does not define requirements for MTA and therefore it has not been evaluated.
Non FIPS-CC mode of operation	The TOE runs in FIPS-CC mode of operation. Non FIPS-CC mode of operation is excluded from the evaluation configuration.
Any features not associated with SFRs in claimed [cPPND]	[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.

3 Security Problem Definition

This ST includes by reference the Security Problem Definition, composed of threats, assumptions, and organizational security policies from the NDcPP. 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 threat model of the NDcPP is designed to protect against the following:

- Unauthorized or insecure communications
- Invalid updates
- Undetected activity
- Unauthorized administrators
- Device failures

The NDcPP defines several assumptions that only apply to the TOE in certain circumstances; within the context of this ST, the A.COMPONENTS_RUNNING assumption does not apply because the TOE is a standalone device, and the A.VS_TRUSTED_ADMINISTRATOR, A.VS_REGULAR_UPDATES, and A.VS_ISOLATION assumptions all apply because the TOE has a virtual network device model.

4 Security Objectives

Like the Security Problem Definition, this ST includes by reference the security objectives defined in the NDcPP. This includes security objectives for the TOE's operational environment.

The NDcPP defines several objectives that only apply to the TOE in certain circumstances; within the context of this ST, the OE.COMPONENTS_RUNNING objective does not apply because the TOE is a standalone device, and the OE.VM_CONFIGURATION objective does apply because the TOE includes a virtual network device.

5 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 Target of Evaluation (TOE) and to scope the evaluation effort.

The SFRs have all been drawn from the following Protection Profile (PP):

- *collaborative Protection Profile for Network Devices*, Version 2.2e, March 23, 2020

As a result, any selection, assignment, or refinement operations already performed by that PP on the claimed SFRs are not identified here (i.e., they are not formatted in accordance with the conventions specified in section 1.3 of this ST). Formatting conventions are only applied on SFR text that was chosen at the ST author's discretion.

5.1 Extended Requirements

All of the extended requirements in this ST have been drawn from the NDcPP. These documents define the following extended SAR and extended SFRs; since they have not been redefined in this ST, the NDcPP should be consulted for more information regarding these extensions to CC Part 2.

- FAU_STG_EXT.1
- FCS_HTTPS_EXT.1
- FCS_RBG_EXT.1
- FCS_SSHS_EXT.1
- FCS_TLSC_EXT.1
- FCS_TLSC_EXT.2
- FCS_TLSS_EXT.1
- FIA_PMG_EXT.1
- FIA_X509_EXT.1/Rev
- FIA_X509_EXT.2
- FIA_X509_EXT.3
- FIA_UAU_EXT.2
- FIA_UIA_EXT.1
- FPT_APW_EXT.1
- FPT_SKP_EXT.1
- FPT_STM_EXT.1
- FPT_TST_EXT.1
- FPT_TUD_EXT.1
- FTA_SSL_EXT.1

5.2 TOE Security Functional Requirements

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

Table 3: TOE Security Functional Components

Requirement Class	Requirement Component
FAU: Security Audit	FAU_GEN.1 Audit Data Generation
	FAU_GEN.2 User Identity Association

Requirement Class	Requirement Component
	FAU_STG_EXT.1 Protected Audit Event Storage
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/Hash Cryptographic Operation (Hash Algorithm)
	FCS_COP.1/KeyedHash Cryptographic Operation (Keyed Hash Algorithm)
	FCS_COP.1/SigGen Cryptographic Operation (Signature Generation and Verification)
	FCS_HTTPS_EXT.1 HTTPS Protocol
	FCS_RBG_EXT.1 Random Bit Generation
	FCS_SSHS_EXT.1 SSH Server Protocol
	FCS_TLSC_EXT.1 TLS Client Protocol Without Mutual Authentication
	FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication
	FCS_TLSS_EXT.1 TLS Server Protocol Without Mutual Authentication
FIA: Identification and authentication	FIA_AFL.1 Authentication Failure Management
	FIA_PMG_EXT.1 Password Management
	FIA_UAU.7 Protected Authentication Feedback
	FIA_UAU_EXT.2 Password-Based Authentication Mechanism
	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/Functions Management of Security Functions Behaviour
	FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour
	FMT_MTD.1/CoreData Management of TSF Data
	FMT_MTD.1/CryptoKeys Management of TSF Data
	FMT_SMF.1 Specification of Management Functions
	FMT_SMR.2 Restrictions on Security Roles
FPT: Protection of the TSF	FPT_APW_EXT.1 Protection of Administrator Passwords
	FPT_SKP_EXT.1 Protection of TSF Data (for reading 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.3 TSF-Initiated Termination
	FTA_SSL.4 User-Initiated Termination

Requirement Class	Requirement Component
	FTA_SSL_EXT.1 TSF-Initiated Session Locking
	FTA_TAB.1 Default TOE Access Banners
FTP: Trusted Path/Channels	FTP_ITC.1 Inter-TSF Trusted Channel
	FTP_TRP.1/Admin Trusted Path

5.2.1 Security Audit (FAU)

5.2.1.1 FAU_GEN.1 Audit Data Generation

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

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

d) Specifically defined auditable events listed in Table **4**.

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 **4**.

Table 4: Auditable Events

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

Requirement	Auditable Events	Additional Audit Record Contents
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session	Reason for failure
FCS_RBG_EXT.1	None.	None.
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
FCS_TLSC_EXT.1	Failure to establish a TLS session	Reason for failure
FCS_TLSC_EXT.2	None.	None.
FCS_TLSS_EXT.1	Failure to establish a TLS session	Reason for failure
FIA_AFL.1	Unsuccessful login attempts limit is met or exceeded.	Origin of the attempt (e.g., IP address).
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None
FIA_X509_EXT.1 /Rev	Unsuccessful attempt to validate a certificate	Reason for failure of certificate validation.
	Any addition, replacement, or removal of trust anchors in the TOE's trust store	Identification of certificates added, replaced, or removed as trust anchor in the TOE's trust store.
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Functions	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_APW_EXT.1	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_STM_EXT.1	Discontinuous changes to time - either Administrator actuated or changed via an automated process.	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).
FPT_TST_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FTA_SSL_EXT.1 (if "terminate the session" is selected)	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions	None.

5.2.1.2 FAU_GEN.2 User Identity Association

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

5.2.1.3 FAU_STG_EXT.1 Protected Audit Event Storage

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

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

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

FAU_STG_EXT.1.3 The TSF shall [overwrite previous audit records according to the following rule: [*the oldest audit records are overwritten*]] when the local storage space for audit data is full.

5.2.2 Cryptographic Support (FCS)

5.2.2.1 FCS_CKM.1 Cryptographic Key Generation

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

- RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;

- ECC schemes using 'NIST curves' [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4
- FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1
- 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]

].

5.2.2.2 FCS_CKM.2 Cryptographic Key Establishment[1]

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"]
- Finite field-based key establishment schemes that meet the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";
- 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]].

5.2.2.3 FCS_CKM.4 Cryptographic Key Destruction

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

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

that meets the following: No Standard.

[1] Modified by TD0580 and TD0581

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).*

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

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

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

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

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

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

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

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

- RSA Digital Signature Algorithm and cryptographic key sizes (modulus) *[2048 bits, 3072 bits]*
- Elliptic Curve Digital Signature Algorithm and cryptographic key sizes *[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 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3],
- For ECDSA schemes: FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Section 6 and Appendix D, Implementing “NIST curves” [P-256, P-384, P-521]; ISO/IEC 14888-3, Section 6.4.

5.2.2.8 FCS_HTTPS_EXT.1 HTTPS Protocol

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

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

FCS_HTTPS_EXT.1.3 If a peer certificate is presented, the TSF shall [not establish the connection] if the peer certificate is deemed invalid.

5.2.2.9 FCS_RBG_EXT.1 Random Bit Generation Services

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

5.2.2.10 FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol in accordance with: RFCs 4251, 4252, 4253, 4254, [4256, 5647, 6668].

FCS_SSHS_EXT.1.2[2] The TSF shall ensure that the SSH protocol implementation supports the following user authentication methods as described in RFC 4252: public key-based, [password-based].

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

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

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH public-key based authentication implementation uses [rsa-sha2-512, ecdsa-sha2-nistp384] as its public key algorithm(s) and rejects all other public key algorithms.

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

FCS_SSHS_EXT.1.7 The TSF shall ensure that [diffie-hellman-group16-sha512, ecdh-sha2-nistp384] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and each encryption key is

[2] Modified by TD0631

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

5.2.2.11 FCS_TLSC_EXT.1 TLS Client Protocol Without Mutual Authentication

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

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

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 SAN, IPv6 address in the SAN].

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

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

5.2.2.12 FCS_TLSC_EXT.2 TLS Client Support for Mutual Authentication

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

5.2.2.13 FCS_TLSS_EXT.1 TLS Server Protocol Without Mutual Authentication

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

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,

- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

and no other ciphersuites.

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

FCS_TLSS_EXT.1.3 The TSF shall perform key establishment for TLS using [Diffie-Hellman parameters with size *[2048 bits]*, ECDHE curves *[secp256r1, secp384r1, secp521r1]*].

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

5.2.3 Identification and Authentication (FIA)

5.2.3.1 FIA_AFL.1 Authentication Failure Management

FIA_AFL.1.1 The TSF shall detect when an Administrator configurable positive integer within [*3-5*] unsuccessful authentication attempts occur related to Administrators attempting to authenticate remotely using a password.

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

5.2.3.2 FIA_PMG_EXT.1 Password Management

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: [“!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”];
b) Minimum password length shall be configurable to between [*8*] and [*15*] characters.

5.2.3.3 FIA_UAU.7 Protected Authentication Feedback

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

5.2.3.4 FIA_UAU_EXT.2 Password-Based Authentication Mechanism

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

5.2.3.5 FIA_UIA_EXT.1 User Identification and Authentication

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

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

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

5.2.3.6 FIA_X509_EXT.1/Rev X.509 Certificate Validation

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

- RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The certification path must terminate with a trusted CA certificate designated as a trust anchor.
- The TSF shall validate a certification path by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The TSF shall validate the revocation status of the certificate using [the Online Certificate Status Protocol (OCSP) as specified in RFC 6960, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3]
- The TSF shall validate the extendedKeyUsage field according to the following rules:
  - Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.
  - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
  - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
  - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

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

5.2.3.7 FIA_X509_EXT.2 X.509 Certificate Authentication

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

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

5.2.3.8 FIA_X509_EXT.3 X.509 Certificate Requests

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

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

5.2.4 Security Management (FMT)

5.2.4.1 FMT_MOF.1/Functions Management of Security Functions Behaviour

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

5.2.4.2 FMT_MOF.1/ManualUpdate Management of Security Functions Behaviour

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

5.2.4.3 FMT_MTD.1/CoreData Management of TSF Data

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

5.2.4.4 FMT_MTD.1/CryptoKeys Management of TSF Data

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

5.2.4.5 FMT_SMF.1 Specification of Management Functions

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using* [*digital signature*] *capability prior to installing those updates;*
- *Ability to configure the authentication failure parameters for FIA_AFL.1; [*
- *Ability to modify the behaviour of the transmission of audit data to an external IT entity;*

- *Ability to manage the cryptographic keys;*
- *Ability to manage the trusted public keys database*[3]
- *Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchors;*
- *Ability to import X.509v3 certificates to the TOE's trust store;*
- *Ability to set the time which is used for time-stamps.*

].

5.2.4.6 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- Security Administrator.

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

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

are satisfied.

5.2.5 Protection of the TSF (FPT)

5.2.5.1 FPT_APW_EXT.1 Protection of Administrator Passwords

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

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

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

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

5.2.5.3 FPT_STM_EXT.1[4] Reliable Time Stamps

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

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

5.2.5.4 FPT_TST_EXT.1 TSF Testing

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

[3] Added per TD0631

[4] Modified by TD0632

operation of the TSF: [*Configuration file integrity test, Firmware integrity test, cryptographic known answer tests, SP 800-90A health tests, RNG known answer test, conditional tests*].

5.2.5.5 FPT_TUD_EXT.1 Trusted Update

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

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

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

5.2.6 TOE Access (FTA)

5.2.6.1 FTA_SSL.3 TSF-Initiated Termination

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

5.2.6.2 FTA_SSL.4 User-Initiated Termination

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

5.2.6.3 FTA_SSL_EXT.1 TSF-Initiated Session Locking

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

5.2.6.4 FTA_TAB.1 Default TOE Access Banners

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

5.2.7 Trusted Path/Channels (FTP)

5.2.7.1 FTP_ITC.1 Inter-TSF Trusted Channel

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

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

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

5.2.7.2 FTP_TRP.1/Admin Trusted Path

FTP_TRP.1.1/Admin The TSF shall be capable of using [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.

5.3 TOE Security Assurance Requirements

The security assurance requirements for the TOE are included by reference to the NDcPP.

Table 5: Assurance Components

Requirement Class	Requirement Component
ADV: Development	ADV_FSP.1 Basic Functional Specification
AGD: Guidance Documentation	AGD_OPE.1 Operational User Guidance
	AGD_PRE.1 Preparative Procedures
ALC: Life-cycle Support	ALC_CMC.1 Labeling of the TOE
	ALC_CMS.1 TOE CM Coverage
ASE: Security Target	ASE_CCL.1 Conformance Claims
	ASE_ECD.1 Extended Components Definition
	ASE_INT.1 ST Introduction
	ASE_OBJ.1 Security Objectives for the Operational Environment
	ASE_REQ.1 Stated Security Requirements
	ASE_SPD.1 Security Problem Definition
	ASE_TSS.1 TOE Summary Specification
ATE: Tests	ATE_IND.1 Independent Testing – Conformance
AVA: Vulnerability Assessment	AVA_VAN.1 Vulnerability Survey

All SARs required by the NDcPP will apply to the entire TOE. The evaluation activities specified in the NDcPP apply to the entire TOE evaluation, including any changes made to them by subsequent NIAP Technical Decisions as summarized in section 1.2 above.

6 TOE Summary Specification

This chapter describes the security functions of the TOE:

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

6.1 Security Audit

The TOE is a standalone device that generates audit records residing in its local storage. The TOE generates audit records of security-relevant management functions on both its local and remote interfaces. The audit records that are generated include startup and shutdown of the TOE (the audit functions are not enabled or disabled separately), all administrative actions, and the specific events identified in Table 4 above. For all auditable events that involve a user (e.g. authentication and administration events), the user identity is captured in the audit record. For cryptographic key related events, the associated certificate label is recorded in the audit record or for SSH, the generation of the key is recorded along with associated identifying information: admin user and ip address. All audit records also include date and time of the event, type of event, subject identity, and the outcome of the event where appropriate.

The TOE is capable of simultaneously logging the audit messages both locally and remotely. Security relevant audit records are stored in the klog file (default location) and can be accessed from the ***Log*** submenu of the ***Monitor*** tab. These are referred to as System Events or ***kevent*** log types. Only authorized administrators may delete log files or view log records, and no capability to modify the records is provided.

The default settings for the TOE in FIPS-CC mode, specify the TOE will log locally and will overwrite the oldest audit logs upon hitting the threshold of memory capacity. This threshold is hit when the log partition of the local disk is almost full, meaning that less than 5 percent of the disk space or 1.5 GB, whichever is smaller, is left. 80% of the appliance disk capacity is reserved for local audit log storage. The capacity of the largest appliance is 20 TB. This translates to a maximum log file size capacity of 16 TB.

The TOE has configurable options for the remote storage of the audit events. These events are sent to one or more configured audit servers, in real-time, simultaneously with the audit records that are written locally. In the evaluated configuration, the TOE uses a FAZ as its external audit server. Communications with this server use TLS.

6.2 Cryptographic Support

The TOE implements cryptographic functionality using FortiMail Cryptographic Library 7.4. The TOE uses the cryptographic functions for outbound TLS connections to environmental audit servers and for the remote administrative SSH and GUI connections.

Table 6 below identifies the algorithm certificates that apply to all cryptographic libraries.

Table 6: Validated Algorithm Implementations

Functions	Standards	Certificates
Asymmetric key generation (FCS_CKM.1)		
RSA Schemes (2048, 3072-bit)	FIPS PUB 186-4, “Digital Signature Standard (DSS)”, Appendix B.3	A5164 A5175

Functions	Standards	Certificates
ECC Schemes (ECDSA P-256, P-384, P-521 curves)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4	A5164 A5175
FFC Schemes (2048-bit DSA)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1	A5164 A5175
FFC Schemes ('safe-prime' groups (2048-bit MODP, 4096-bit MODP)	NIST SP 800-56A Revision 3; RFC 3526	A5164 A5175
Key Establishment (FCS_CKM.2)		
Elliptic curve-based scheme (ECDSA) P-256, P-384, P-521	NIST Special Publication 800-56A Revision 3	A5164 A5175
Finite field-based scheme: FFC Schemes (FB/FC)	NIST Special Publication 800-56A Revision 3, RFC 3526	A5164 A5175
FFC Schemes using 'safe-prime' groups (2048-bit MODP, 4096 bit MODP)	NIST Special Publication 800-56A Revision 3, RFC 3526	A5164 A5175
Encryption/Decryption (FCS_COP.1/DataEncryption)		
AES in CBC mode (128, 256 bits)	AES as specified in ISO 18033-3 CBC as specified in ISO 10116	A5164 A5175
AES in GCM mode (128, 256 bits)	AES as specified in ISO 18033-3 GCM as specified in ISO 19772	A5164 A5175
Cryptographic signature services (Signature Generation and Verification) (FCS_COP.1/SigGen)		
RSA Digital Signature Algorithm (rDSA) (2048, 3072-bit modulus)	FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 5.5, using PKCS #1 v2.1 Signature Schemes RSASSA-PSS and/or RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3	A5164 A5175
ECDSA (P-256, P-384, P-521)	ECDSA schemes: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Section 6 and Appendix D, Implementing "NIST curves"	A5164 A5175
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	A5164 A5175
Keyed-hash message authentication (FCS_COP.1/KeyedHash)		

Functions	Standards	Certificates
HMAC-SHA-1 (key/digest sizes 160 bits) HMAC-SHA-256 (key/digest sizes 256 bits) HMAC-SHA-384 (key/digest sizes 384 bits)	ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2	A5164 A5175
Random bit generation (FCS_RBG_EXT.1)		
CTR-DRBG (AES) – 256 bits entropy	ISO/IEC 18031:2011 Table C.1 "Security Strength Table for Hash Functions	A5164 A5175

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

- TLS Server: 2048 and 3072 bits RSA, P-256, P-384, P-521 curves ECC, and MODP 2048 bit safe primes per RFC 3526,
- TLS Client: 2048 and 3072 bits RSA, P-256, P-384, P-521 curves ECC, 2048-bit DSA, and MODP 2048 bit safe primes per RFC 3526,
- X.509 Certificate Requests: 2048 and 3072 bits RSA, and P-256, P-384, P-521 curves ECC key generation
- SSH: 2048 and 3072 bits RSA, P-256, P-384, P-521 ECC curves, and MODP 4096 bit safe primes per RFC 3526

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

- TLS Server: ECDSA (P-256, P-384, P-521) and MODP 2048 bit safe primes per RFC 3526
- TLS Client: ECDSA (P-256, P-384, P-521), FFC (FB/FC) using 2048-bit DSA, and DH group 14 (MODP 2048-bit safe primes) per RFC 3526
- SSH: ECDSA (P-384) and FFC MODP 4096 bit safe primes per RFC 3526

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)
- RSA signature generation and verification (2048, 3072 bit)
    - 2048-bit RSA with SHA-256 is also used for TOE update, digital signature verification for firmware integrity, and self-tests
- ECDSA signature generation and verification (P-256, P-384, P-521)
- SHA-1, SHA-256, SHA-384, SHA-512
    - SHA-256 is also used for administrator password hashing for non-volatile storage, digital signature verification for self-test of configuration integrity and for integrity of firmware update
    - SHA-512 is only used for SSH
- HMAC-SHA-1 (160 bit key/block size, 160 bit output length), HMAC-SHA-256 (256 bit key/block size, 256 bit output length), HMAC-SHA-384 (384 bit key/block size, 384 bit output length)
    - HMAC-SHA-256 is also used for self-test of configuration integrity and for integrity of firmware updates

The TOE uses JitterEntropy 3.4.1 as a raw entropy source collected from CPU execution time jitter. The collected entropy is injected into the Linux kernel /dev/random device using the RNDADDENTROPY ioctl. This noise source provides full entropy to seed the DRBG with 256 bits. The Fortinet FortiMail Cryptographic Library Version 7.4 contains a CTR_DRBG implemented in accordance with ISO/IEC 18031:2011 Deterministic Random Bit Generator (DRBG) based on the AES 256 block cipher in counter mode (CTR_DRBG (AES-256)). Entropy from the noise source is used to seed the DRBG with 256 bits of full entropy. A failure of the entropy source is a blocking event for the cryptographic system and the entropy source is continually monitored for health; this helps ensure that a catastrophic failure of the noise source will halt the operation of the TOE. The TOE uses its DRBG to generate all keys.

The TOE maintains a number of keys and CSPs related to its secure operation (see Table 7). Administrative passwords are stored in the configuration file on the flash drive of the TOE and are hashed using SHA-256 to ensure their confidentiality. Keys can be zeroized either by erasing the boot device or through a factory reset of the TOE.

The following table describes the origin, storage and zeroization of keys as relevant to FCS_CKM.4 and FPT_SKP_EXT.1 provided by the TOE.

Table 7: Key/CSP Storage and zeroization

Key or CSP	Storage	Zeroization Method	Origin
Firmware Update Key	Flash (Plaintext)	Erase boot device (overwritten with zeroes)	Preconfigured
HTTPS/TLS Pre-Master Secret	RAM (Plaintext)	Power cycle or reboot; session terminated (overwritten with zeroes)	Automatic
SSH Server/Host Key	Flash (Plaintext)	Erase boot device (overwritten with zeroes) overwrite with new key	Preconfigured and can be regenerated/modified by the administrator using exec ssh-regen-keys
Firmware Integrity Key	Flash (Plaintext)	Erase boot device (overwritten with zeroes)	Preconfigured
HTTPS/TLS Pre-Master/Master Secrets	RAM	Power cycle or reboot; session terminated (overwritten with zeroes)	Automatic
HTTPS/TLS Server/Host Key	Flash (Plaintext) RAM	Erase boot device (overwritten with zeroes) Power cycle or reboot; session terminated (overwritten with zeroes)	Preconfigured
HTTPS/TLS Session	RAM	Power cycle or reboot; session terminated (overwritten with zeroes)	Automatic

Authentication Key			
HTTPS/TLS Session Encryption Key	RAM	Power cycle or reboot; session terminated (overwritten with zeroes)	Automatic
SSH Session Authentication Key	RAM	Power cycle or reboot; session terminated (overwritten with zeroes)	Automatic
SSH Session Encryption Key	RAM	Power cycle or reboot; session terminated (overwritten with zeroes)	Automatic
Configuration Integrity Key	Flash (Plaintext)	Erase boot device (overwritten with zeroes)	Preconfigured
Configuration Encryption Key	RAM	Power cycle or reboot; (overwritten with zeroes)	Preconfigured
Configuration Backup Key	Flash (Plaintext)	Erase boot device (overwritten with zeroes)	Automatic
User/Admin Passwords	Flash (Hashed)	Factory reset (overwritten with zeroes)	Manual

The TOE stores a number of CSPs in volatile memory during normal operation of the cryptographic modules. These CSPs include plaintext ephemeral keys and copies of the persistent keys described above are loaded into memory during normal operation. The TOE maintains these keys in its volatile memory in to support the TLS and HTTPS connections to the TOE. These CSPs are cleared when the appliance power cycles or reboots. Ephemeral keys are overwritten with a fixed pattern (zeroes) when they are no longer required. Each of the CSPs are protected from unauthorized access via memory management which disallows any memory reads from other processes within the OS ensuring that the CSPs are only available to the calling application.

Plaintext private keys for the purposes of SSH, HTTPS and TLS are maintained on the flash filesystem and are not viewable through the TOE interfaces. When these keys are no longer required the administrator can remove the keys by erasing the boot device.

All keys and CSPs can be zeroized / overwritten by Linux kernel and the Command Line Interface (i.e. execute erase-filesystem and exec ssh-regen-keys). All keys and CSPs can be zeroized by executing the following command from the CLI: execute erase-filesystem. The erase-filesystem command performs a direct overwrite (no intermediate file system APIs are used).

The TOE does not provide any interfaces to view the keys/CSPs.

The TOE's SSH server implements the SSH protocol in accordance with RFCs 4251, 4252, 4253, 4254, 4256, 5647 and 6668. No optional characteristics are supported. The TOE supports SSH public key-based and password-based authentication methods as described in RFC 4252 and as described in RFC 4253, packets greater than 256K bytes in an SSH transport connection are dropped. User public key authentication uses RSA.

For its SSH transport implementation, the TOE uses aes128-cbc, aes256-cbc, aes128-gcm@openssh.com, and aes256-gcm@openssh.com, rejecting all other encryption algorithms. This is hardcoded and not configurable.

The SSH public-key based authentication implementation uses only rsa-sha2-512 and ecdsa-sha2-nistp384 as its public key algorithms and rejects all other public key algorithms. The TOE establishes a user identity when an SSH client presents a public key by verifying that the SSH client's presented public key matches one that is stored within the SSH server's authorized_keys file.

The SSH transport implementation uses hmac-sha1 and hmac-sha2-256 as its data integrity MAC algorithms and rejects all other MAC algorithms. Additionally, when either aes*-gcm@openssh.com algorithm is negotiated as the encryption algorithm, the MAC algorithm field is ignored and GCM is implicitly used as the MAC.

The TOE's SSH implementation uses only diffie-hellman-group16-sha512 and ecdh-sha2-nistp384 for its key exchange methods.

Within SSH connections the same session keys are used for a threshold of no longer than one hour, and each encryption key is used to protect no more than 512 MB of data. After either of the thresholds are reached, the TOE performs a rekey.

The TOE implements TLS and HTTPS protocols for external communications. The TOE supports HTTPS to secure the sessions for remote administration over TLSv1.1 and 1.2. The TOE does not use HTTPS in a client capacity. The TOE's HTTPS protocol complies with RFC 2818 and uses the TLS functionality described below. RFC 2818 (HTTP Over TLS) describes how to use TLS to secure HTTP connections over the Internet. The details of the TSF behavior with regards to the following requirements listed in RFC 2818 are as follows:

- Section 2.1, Connection Initiation:
  - The TSF acts as the HTTPS Server, therefore this section does not pertain to the TSF
- Section 2.2, Connection Closure:
  - The TSF sends TLS closure alert when terminating an HTTPS connection
  - The TSF supports session reuse
  - The TSF meets described behavior; no deviation
- Section 2.2.1, Client Behavior:
  - The TSF acts as the HTTPS Server, therefore this section does not pertain to the TSF
- Section 2.2.2, Server Behavior:
  - The TSF supports TLS session resumption based on session tickets according to RFC 5077
  - The TSF will attempt to initiate an exchange of closure alerts with the client before closing the connection
- Section 2.3, Port Number:
  - The TSF utilizes TCP port 443 to listen for incoming HTTPS connections
- Section 2.4, URI Format:
  - The TSF supports and requires the "https://" URI protocol identifier prefix for incoming HTTPS requests
- Section 3.1, Server Identity:
  - Pertains to client behavior, not applicable to the TOE
- Section 3.2, Client Identity:
  - The TSF does not support mutual authentication with regards to the HTTPS interface

TLS 1.1 or 1.2 is also used for the purposes of protecting the audit logs while in transit to the audit servers. In all cases, an invalid certificate will cause the connection to be aborted.

The TOE implements TLS 1.1 and TLS 1.2 as both a client and a server, rejecting all other TLS/SSL versions. Specifically, enabling FIPS-CC mode (required in evaluated configuration) forces the TOE to use only TLS versions 1.1 or 1.2. All other versions are automatically disabled by default.

The TOE is a TLS client when communicating with external audit servers. Mutual authentication is required and the following cipher suites are supported:

- TLS_DHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492,
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256 as defined in RFC 5246,
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289,
- TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289

The TOE is a TLS server when communicating with remote administrative users accessing the TOEs GUI. Mutual authentication is not supported. The TOE supports the same cipher suites as identified for the TLS client.

The TOE operates in FIPS-CC mode of operation that restricts the cipher suites and algorithms used by HTTPS/TLS to those identified above. The administrator does not need to take any specific actions to ensure compliance once the FIPS-CC mode of operation has been enabled.

In the evaluated configuration, the TOE supports the following parameters for TLS. The TLS client and server uses secp256r1, secp384r1, or secp521r1 when a TLS_ECDHE cipher suite is negotiated. When a TLS_DHE cipher suite is negotiated, a 2048-bit MODP Diffie-Hellman parameter is used. This is the default behavior in FIPS-CC mode and is not configurable.

When acting as a TLS client, the TOE presents a TLS client certificate with the Client Authentication purpose in the extendedKeyUsage filed in support of mutual authentication. The server validates the TOE's TLS client certificate using the presented X.509v3 certificate.

The TOE supports wildcards for DNS names in the SAN and CN. IPv4/IPv6 addresses and DNS names are supported in the SAN and are configured by the administrative user. The TOE validates any presented server certificate in the manner specified by RFC 6125 section 6. Specifically, if the SAN is present, the client uses the IP address or DNS name in the SAN as the reference identifier for the TLS server certificate. When the SAN is not available, the TOE makes use of the FQDN (CN). The CN only supports DNS names. Invalid certificates are rejected. No administrator override exists to re-adjudicate the rejection of an invalid certificate.

The TLS server supports session resumption based on session tickets. Session tickets adhere to the structural format provided in section 4 of RFC 5077. Session tickets are encrypted according to the TLS negotiated symmetric encryption algorithm consistent with FCS_COP.1/DataEncryption. Depending on the negotiated ciphersuite, AES in CBC or GCM mode (128, 256 bits) are used to protect session tickets.

6.3 Identification and Authentication

The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed aside from display of the warning banner to the administrator. Administrative access to the TOE is facilitated by either directly connecting to the appliance console's CLI, remotely to the CLI via SSH, or by remotely connecting to appliance GUI via HTTPS/TLS.

Regardless of the interface at which the administrator interacts, the TOE prompts the user for a credential. The administrator credentials are the same across the interfaces (i.e. the interfaces do not have separate credentials). Only after the administrative user presents the correct authentication credentials will they be granted access to the TOE administrative functionality. Passwords or keys can be used for the SSH connection. A key presented for SSH login must match that for the defined account in the TOE's database. If the asserted identity and password or key cannot be verified, then the login fails and an audit record is generated.

The TOE provides a local password based authentication mechanism. The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters (that include: "!", "@", "#", "$", "%", "^", "&", "*", "(", and ")"). The minimum password length is settable by the Authorized Administrator and can range from 8 to 15 characters.

The TOE provides administrators with the capability to specify a maximum number of authentication attempts between 3 and 5 (default 4) that can be attempted before a user account is locked out from the remote GUI or SSH. The TSF increments the failure counter in non-volatile memory so that the accumulated number of failures is persisted across reboots. The failure counter is per administrator account. In the event the remote administrator is locked out, access is restored by waiting for the lockout period to expire. The lockout period is a configurable value between 1 and 60 minutes, with a default value of 3 minutes. Once the maximum number of attempts has been reached, the account will become locked and inaccessible until an administrator configured period of time has been met. The TOE supports a local interface that is not subject to the authentication failure lockout function and does not lock an administrative user out. Even if an administrator is locked out from a remote interface, they can still login locally. For a virtual TOE, the administrator signs in to ESXi console to launch a local console for each VM. This prevents a situation where no administrator access is available.

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

The TOE performs X.509 certificate validation in support of TLS and TLS/HTTPS communications. All certificates that are presented to the TOE are validated. This includes TLS server certificates where the

TOE acts as a TLS client and the TOE's own TLS server/client certificates when a CA response to a certificate signing request is provided back to the TOE. Specifically, the following validation rules are followed:

- The TSF performs RFC 5280 certificate validation and certification path validation supporting a minimum path length of three certificates.
- The TSF validates that the certification path terminates with a trusted CA certificate designated as a trust anchor.
- The certification path is validated by ensuring that all CA certificates in the certification path contain the basicConstraints extension with the CA flag set to TRUE.
- The extendedKeyUsage field is validated according to the following rules:
    - Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.
    - Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.
    - OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.

Note that FIA_X509_EXT.1/Rev also optionally requires the TSF to verify the extendedKeyUsage field for certificates used for trusted updates and executable code integrity verification. The TOE does not use X.509 certificates for these purposes so this portion of the requirement is trivially satisfied by the TSF.

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. For mutual authentication with external audit server, the TOE chooses the certificate to present to the server based on the client certificate imported into the TOE 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 supports the use of OCSP and CRL as specified in RFC 6960 and RFC 5280 for revocation status checking of TLS certificates. In all cases, if the revocation status of a certificate cannot be determined, the TSF treats it as invalid. Both the leaf certificate and any intermediate CA certificates are checked for TLS.

The TOE also includes the ability to generate a certificate signing request as specified by RFC 2986 to be sent to a CA for issuance of TLS server/client certificates. The certificate signing request generates an RSA or elliptic curve key pair and can include the Common Name, Organization, Organizational Unit, and Country fields. When the signed response is loaded into the TOE, the TSF validates the certificate chain from the root CA and will accept the response as the TLS server certificate if the certificate chain is valid. This is subsequently the certificate that the TOE issues to external TLS clients attempting to connect to it.

6.4 Security Management

The TOE provides management functionality over both local and remote interfaces. The local interface is a dedicated physical port, direct console interface to the TOE's CLI. Local access to a virtual TOE is through the ESXi console via URL. The remote interfaces are an SSH connection to the CLI and a web GUI accessed over TLS/HTTPS. The TOE supports one default fully privileged user account, Admin, which corresponds to the PP defined Security Administrator. The Admin account has full privileges to all administrative functions.

Table 8 below identifies the management functions that are available on each interface. The local management functions available are those available on the console and the functions available to remote admin interfaces are available via SSH and/or HTTPS. All remote functions can be performed via the CLI

and the table notes the specific functions that cannot be performed via the GUI via footnote. All functions are restricted to the Security Administrator.

Table 8: Management Functions by Interface

Function	Local	Remote
Ability to administer the TOE locally	X	
Ability to administer the TOE remotely		X
Ability to configure the access banner	X	X
Ability to configure the session inactivity time before session termination	X	X
Ability to update the TOE and verify the updates using digital signature capability prior to installing those updates	X	X
Ability to configure authentication failure parameters for FIA_AFL.1	X	X[5]
Ability to modify the behavior of the transmission of audit data to an external IT entity	X	X
Ability to manage the cryptographic keys	X	X
Ability to manage the trusted public keys database	X	X[5]
Ability to set the time which is used for time-stamps	X	X
Ability to manage the TOE's trust store and designate X.509v3 certificates as trust anchors	X	X
Ability to import X.509v3 certificates to the TOE's trust store	X	X

The Security Administrator is able to configure transmission of audit data to an external audit server.

The Security Administrator has the ability to manage cryptographic keys to manipulate (add, remove) the certificates (and keys) that reside in the TOE's trust store and to generate certificate signing requests, which include key pairs.

The Security Administrator can manage the trusted public keys database containing SSH keys.

6.5 Protection of the TSF

The TOE implements various self-protection mechanisms for its functions and data.

The TSF stores all Administrative password data in an obfuscated format. Specifically, the data is stored as a SHA-256 hash. The TSF stores its private key on its local file system when the key pair for the CSR is first generated. When the signed certificate response is received by the TOE, the certificate and private key are stored in a secure directory that is not accessible to administrators. The TOE stores its SSH private host key in plaintext in flash and does not provide any interfaces to view the key. The TLS server supports session resumption based on session tickets. Session tickets adhere to the structural format provided in section 4 of RFC 5077. Session tickets are encrypted according to the TLS negotiated symmetric encryption algorithm consistent with FCS_COP.1/DataEncryption. Depending on the negotiated ciphersuite, AES in CBC or GCM mode (128, 256 bits) are used to protect session tickets.

For both credential and non-public key data, no administrative interface exists to read this data.

[5] This function is only available locally or remotely using SSH to the CLI.

The TOE is a hardware appliance or a virtual appliance image installed on a hardware appliance that includes a hardware-based real-time clock to ensure that reliable time information is available. The TOE's real-time clock stores the system time and date information. The TOE's embedded OS manages the clock and exposes administrator clock-related functions. The clock is used for audit record time stamps, measuring session activity for termination, timing of lockout in FIA_AFL.1, and for cryptographic operations based on time/date (e.g. validation of X.509 certificates). The Administrator can configure the system time and date of the FortiMail unit using the CLI ***system time manual*** command or from the GUI, using the command: ***System > Configuration > Time.***

The TSF provides a set of self-tests run during initial start-up. During a normal boot-up sequence the TOE administrator can see on the local console the following types of tests:

- Configuration file integrity test: The configuration file integrity test is run automatically at startup. A HMAC SHA-256 hash of the configuration file is compared to the stored pre-computed value and confirms that the configuration information has not been modified since last start.
- Firmware integrity test: The Firmware Integrity Test is run automatically whenever the system images is loaded and confirms through use of RSA 2048-bit and SHA-256 digital signature verification that the image file that's about to be loaded was properly signed and has maintained its integrity since being signed.
- Cryptographic (AES, DHE, SHA, ECDHE, ECDSA, and RSA) known answer tests: For each algorithm, the implementation is fed known plaintext data and a known key (when appropriate). These values are used to generate a value. This value is compared to a known value to verify that the implementation is operating correctly.
- SP 800-90A health tests: For these tests, each of the health tests in defined SP 800-90A are executed.
- RNG known answer test: For this test, known seed values are provided to the DRBG implementation. The DRBG uses these values to generate random bits. These random bits are compared to known random bits to ensure that the DRBG is operating correctly.

The results of the startup self-tests are displayed on the console during the startup process. Indication of successful tests would appear as follows: `Running <test>... passed`; while completion of all self-tests is indicated by: `Self-tests passed.`

The TSF executes the following conditional tests when the related service is invoked:

- Cryptographic (AES, DHE, SHA, KAS, ECDSA sign/verify, and RSA sign/verify) known answer tests: For each algorithm, the implementation is fed known plaintext data and a known key (when appropriate). These values are used to generate a value. This value is compared to a known value to verify that the implementation is operating correctly.
- SP 800-90A health tests: For these tests, each of the health tests in defined SP 800-90A are executed.
- RNG known answer test: For this test, known seed values are provided to the DRBG implementation. The DRBG uses these values to generate random bits. These random bits are compared to known random bits to ensure that the DRBG is operating correctly.

In the event that any of the self-tests or conditional tests fail, the module logs a "error indicator" for the specific test(s) and enters an error state as shown by the following console output:

```
Self-tests failed
Entering error mode...
The system is going down NOW !!
The system is halted.
```

The TOE's Error Mode allows the TOE to enter into a secure state where all data output and cryptographic services are inhibited. The unit shuts down all interfaces including the console and blocks traffic. To resume normal FIPS-CC mode operation, power cycle the unit. If the self-tests pass after the reboot, the unit will resume normal FIPS-CC operation. If a self-test continues to fail after rebooting, there is likely a serious firmware or hardware problem and the unit should be removed from the network until the problem is solved.

These self-tests are sufficient to ensure that the image and configuration file are operating in a known good state, TSF executable code has appropriately not been modified, and that the cryptographic functionality has not been tampered with or degraded, either through a compromise of the cryptographic functionality itself or through a degradation of any hardware components on which this functionality relies. There is no situation in which an administrator may unwittingly be using a modified TOE or may be using the TOE to transmit sensitive data using a degraded cryptographic implementation.

The TOE supports firmware updates. Only one version of the TOE firmware is loaded on disk at any one time. The TOE has a manual update mechanism. The TOE protects itself during updates through the use of a cryptographic signature. The update process is performed as follows. The administrator downloads the TOE to their workstation from https://support.fortinet.com. The administrator will then copy the file to the TOE using: HTTPS web interface, SSH, or direct connection to the console port using a RJ-45 to DB-9 serial cable or a nullmodem cable. Once the update is uploaded to the TOE, a 2048 bit RSA signature is verified for any TOE firmware build (builds include patches) to verify the update is valid. The signature is compared to a known key value stored on the TOE and pre-configured into the firmware image. After image file is uploaded and its integrity is checked using digital signature, a reboot occurs and the image will be executed. The current running version of the firmware can be queried from the CLI using the command: get system status. The version line of the status display shows the FortiMail model number, firmware version, build number and date. For example: Version: FortiMail-2000E v6.0.1,build0102,180627. The firmware version can also be queried from the Web UI's Dashboard>Status tab

Before proceeding with a TOE update via the GUI or CLI, the following automated process is followed when in the evaluated mode of operation:

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

If the firmware load test fails, the error message displayed is "File is not an update file." Otherwise the TOE displays "upgrade successful" and reboots. An administrator may query the current version of the TOE through the CLI or web interface.

6.6 TOE Access

The TOE enforces access restrictions on its local and administrative interfaces. A Security Administrator can configure maximum inactivity times for administrative sessions of 1-480 minutes through the TOE GUI and CLI interfaces. A configured inactivity period will be applied to both local and remote sessions in the same manner. When the interface has been idle for more than the configured period of time, the session will be terminated and will require authentication to establish a new session.

Each interface also includes mechanisms for the Security Administrator on each to voluntarily terminate their own session. On the GUI, this is accomplished using a logout button at the top-righthand side where there is UI to sign admin out, or using the 'exit' command on CLI.

Both local and remote interfaces display a configurable pre-authentication warning banner that can be used to advise Security Administrators of appropriate usage of the system. The custom banner messages are configured using the CLI command "config customized message > edit login-disclaimer" or from the GUI: System > Customization > Custom Message. Configuration from either of these interfaces displays the same configured banner message prior to logging in, and for all interfaces: GUI, SSH, and Console.

6.7 Trusted Path/Channels

The TOE supports communications with external audit servers. These connections are protected via a TLS connection where the TOE is a TLS client and mutual authentication is supported. This protects the data from modification and disclosure. TLS provides assured identification of the non-TSF endpoint by validating X.509 certificates. The TOE retains a trusted store of certificate authorities which it uses to verify digital signatures on those non-TSF certificates. The TOE is responsible for initiating the trusted channel with the external trusted IT entities.

All remote administrative communications take place over a secure encrypted session. Remote CLI sessions occur over an SSH connection. Remote GUI connections take place over a TLS/HTTPS connection. Sessions are encrypted using AES encryption and uses HMACs to protect integrity. The remote administrators can initiate SSH and TLS communications with the TOE. The TOE is the SSH/TLS server and mutual authentication is not supported.

7 Protection Profile Claims

This ST is conformant to the *collaborative Protection Profile for Network Devices, Version 2.2e, 23 March 2020* (NDcPP).

As explained in section 2.6, Excluded Functionality

The TSF is being evaluated as a network device and therefore the email security system providing multi-layered protection against blended threats comprised of spam, viruses, worms and spyware capabilities was not evaluated. 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 2: Excluded Functionality

Feature	Description
Email security system (antivirus, antispam, content filtering, email routing and email archiving)	FotiMail provides an email security system, multi-layered protection against blended threats comprised of spam, viruses, worms and spyware functions. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Inbound filtering engine	FortiMail's inbound filtering engine blocks spam and malware before it can clog networks and affect users. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Dynamic and static user-blocking	FortiMail's dynamic and static user-blocking provides granular control over all email policies and users. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Predefined or customized dictionaries	FortiMail's predefined or customized dictionaries prevent accidental and intentional loss of confidential data. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Identity-Based Encryption (IBE), S/MIME, TLS email encryption	Secure content delivery is enforced with FortiMail's Identity-Based Encryption (IBE), S/MIME, or TLS email encryption options. The [NDcPP] does not define requirements for these functions and therefore they have not been evaluated.
Mail Transfer Agent (MTA)	The [NDcPP] does not define requirements for MTA and therefore it has not been evaluated.

Feature	Description
Non FIPS-CC mode of operation	The TOE runs in FIPS-CC mode of operation. Non FIPS-CC mode of operation is excluded from the evaluation configuration.
Any features not associated with SFRs in claimed [cPPND]	[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.

Security Problem Definition, the Security Problem Definition of the NDcPP has been included by reference into this ST.

As explained in section 4, Security Objectives, the Security Objectives of the NDcPP has been included by reference into this ST.

All claimed SFRs are defined in the NDcPP. All mandatory SFRs are claimed. Selection-based SFR claims are consistent with the selections made in the mandatory SFRs that prompt their inclusion. Some optional SFR claims are made at the TOE developer's discretion.

8 Rationale

This Security Target includes by reference the NDcPP Security Problem Definition, Security Objectives, and Security Assurance Requirements. The Security Target does not add, remove, or modify any of these items. Security Functional Requirements have been reproduced with the Protection Profile operations completed. All selections, assignments, and refinements made on the claimed Security Functional Requirements have been performed in a manner that is consistent with what is permitted by the NDcPP. The proper set of selection-based requirements have been claimed based on the selections made in the mandatory requirements. Consequently, the claims made by this Security Target are sufficient to address the TOE's security problem. Rationale for the sufficiency of the TOE Summary Specification is provided below.

8.1 TOE Summary Specification Rationale

This section in conjunction with Section 6, the TOE Summary Specification, provides evidence that the security functions meet the TOE security requirements. Each description includes rationale indicating which requirements the corresponding security functions satisfy. The combined security functions work together to satisfy all of the security requirements. The security functions described in Section 6 are necessary for the TSF to enforce the required security functionality. Table 9 demonstrates the relationship between security requirements and functions.

Table 9: 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_EXT.1	X						
FCS_CKM.1		X					
FCS_CKM.2		X					
FCS_CKM.4		X					
FCS_COP.1/DataEncryption		X					
FCS_COP.1/Hash		X					
FCS_COP.1/KeyedHash		X					
FCS_COP.1/SigGen		X					
FCS_HTTPS_EXT.1		X					
FCS_RBG_EXT.1		X					
FCS_SSHS_EXT.1		X					
FCS_TLSC_EXT.1		X					
FCS_TLSC_EXT.2		X					

	Security Audit	Cryptographic Support	Identification and Authentication	Security Management	Protection of the TSF	TOE Access	Trusted Path/Channels
FCS_TLSS_EXT.1		X					
FIA_AFL.1			X				
FIA_PMG_EXT.1			X				
FIA_X509_EXT.1/Rev			X				
FIA_X509_EXT.2			X				
FIA_X509_EXT.3			X				
FIA_UIA_EXT.1			X				
FIA_UAU.7			X				
FIA_UAU_EXT.2			X				
FMT_MOF.1/Functions				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_STM_EXT.1					X		
FPT_TST_EXT.1					X		
FPT_TUD_EXT.1					X		
FTA_SSL.3						X	
FTA_SSL.4						X	
FTA_SSL_EXT.1						X	
FTA_TAB.1						X	
FTP_ITC.1							X
FTP_TRP.1/Admin							X