FortiMail™ Appliances Security Target

Document Version: 1.13
Date: January 12, 2016

Prepared For:

Fortinet, Inc.

899 Kifer Rd
Sunnyvale, CA 94086
www.fortinet.com

Prepared By:

Common Criteria Consulting LLC

15804 Laughlin Ln
Silver Spring, MD 20906 USA
www.consulting-cc.com

Revision History

Ver #	Description of changes	Modified by	Date
1.0	Initial version	CCC	June 4, 2014
1.1	Incorporated vendor comments	CCC	June 6, 2014
1.2	Updated TLS info and corrected typos	CCC	June 10, 2014
1.3	Added support for LDAP authentication server and addressed lab ORs	CCC	June 18, 2014
1.4	Addressed lab issues	CCC	June 24, 2014
1.5	Clarified the evaluated configuration	CCC	August 19, 2014
1.6	Addressed vendor and scheme issues	CCC	October 31, 2014
1.7	Addressed certifier issues	CCC	December 3, 2014
1.8	Modified the evaluated appliance models	CCC	May 8, 2015
1.9	Removed SSH, updated TLS cipher suites, name change for the Fortinet Entropy Token	CCC	July 9, 2015
1.10	Clarified CLI access	CCC	July 15, 2015
1.11	Final TOE version update, removed LDAP, updated the list of models not included in the evaluation, addressed scheme comments	CCC	November 18, 2015
1.12	Addressed scheme comments	CCC	December 18, 2015
1.13	Clarifications concerning cryptographic validations	CCC	January 12, 2016

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1 Introduction

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

1.1 ST Reference

ST Title	FortiMail™ Appliances Security Target
ST Revision	1.13
ST Publication Date	January 12, 2016
ST Author	Common Criteria Consulting LLC

1.2 Target of Evaluation Reference

TOE Developer	Fortinet, Inc
TOE Name	FortiMail™ Appliances
TOE Version	FortiMail firmware V5.2.6 (build 460) and Fortinet Entropy Token V1

1.3 Conventions

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

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

1.4 TOE Overview

FortiMail appliances are specialized email security systems that provide 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. Its outbound inspection technology prevents other antispam gateways from blacklisting users by blocking outbound spam and malware, including mobile traffic. 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. To ensure up to date email protection, FortiMail supports Fortinet's FortiGuard™ antispam and antivirus security subscription services.

Administration of the system may be performed locally through the Command Line Interface (CLI) using an administrator console or remotely via a network management station through the FortiMail Web-based manager (using HTTPS). The administrator accesses the CLI via terminal emulation software (e.g. Hyperterm) on a computer co-located with the appliance. This computer is connected to the appliance via a serial cable. Access to the FortiMail administrative functions including audit data is restricted to authenticated Administrators. Administrator authentication is performed by the appliance.

FortiMail supports two high availability modes. Config-only mode provides load balancing and allows up to 25 FortiMail units to share a common configuration, but operate as separate FortiMail units. In Active-passive mode a second (passive) FortiMail unit can be configured as a failover device if the primary (active) FortiMail unit fails. All data from the active unit, except for the Bayesian database, is duplicated to the passive unit.

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.

When operating in gateway mode, FortiMail acts as a Mail Transfer Agent (MTA), also known as an email gateway or relay. 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. 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, all of the system's interfaces are on the same IP subnet and the FortiMail unit effectively acts as a bridge. In transparent mode, the FortiMail system must be physically inline between the protected email server and all SMTP clients — unlike gateway mode. Email clients cannot be configured to route email directly to the FortiMail system, so it must be physically placed where it can intercept the connection.

Fortinet Entropy Token is a USB-based cryptographic support processor that is an option for FortiMail, and is required in the evaluated configuration. For this TOE, Fortinet Entropy Token is used as an entropy source only.

1.5 TOE Description

The TOE is the referenced network appliances as detailed in section 1.5.1 together with the Fortinet Entropy Token. The following figure shows the TOE in a representative deployment.

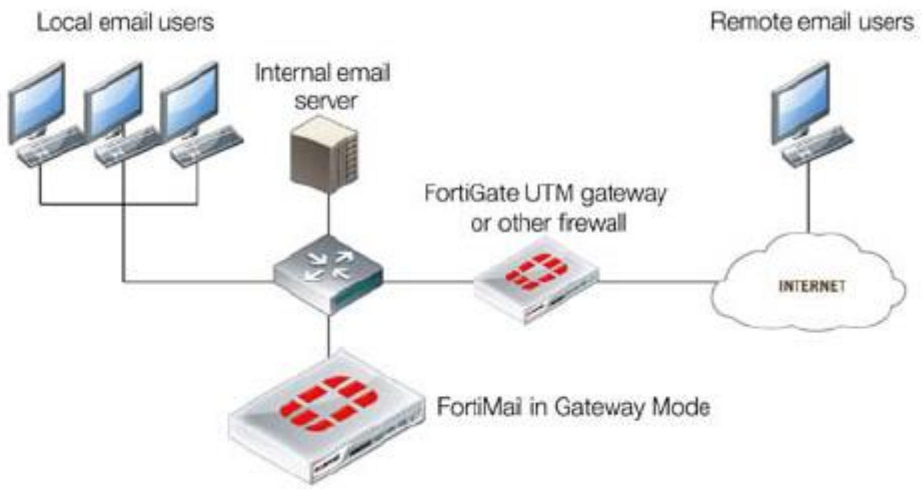

Figure 1 – TOE Representative Deployment

The TOE has extensive logging capabilities. These include administrative actions and logging tampering or misuse of the trusted cryptographic channels. These audit logs are capable of being exported to an external audit server over a cryptographically protected channel for further analysis and inspection, such as by the FortiAnalyzer™ suite of products.

The TOE implements NIST approved cryptography, validated through CAVP. This cryptography is used to secure communications to trusted administrators and to secure generated audit logs in transit to an offsite audit server for additional inspection. The TOE allows for configuration of FortiAnalyzer as a destination for audit logs.

User administration sessions are secured over HTTPS using validated cryptography. To ensure cryptographically strong random number generation the TOE has been equipped with a dedicated USB hardware noise source (Fortinet Entropy Token) which provides entropy collected from the ambient environment in which the product operates. This noise source is continually monitored for its ongoing health and proper operation. The entropy token has been reviewed and approved by the evaluators and certification bodies.

The TOE also offers the ability to verify (through cryptographic signatures) that product updates are valid, and will reject any updates without the appropriate Fortinet signature. The TOE will ensure during boot up that the health of the TOE has not been compromised.

1.5.1 Physical Boundary

The physical scope of the TOE includes all components of the appliance hardware and firmware as well as a Fortinet Entropy Token (part number FTR-ENT1) to provide the hardware noise source. Fortinet Entropy Token is connected to the appliance via a USB port.

Each FortiMail appliance (together with Fortinet Entropy Token) is a stand-alone unit that does not require supporting hardware. The FortiMail unit consists of custom hardware and firmware, including the following major components: firmware, processor, memory, disk storage and I/O interfaces. Each of the appliances provides a local interface and network interfaces for administrator access; the network interfaces are used for email exchanges as well. All of the appliances provide the same security functionality, differing only in the message processing capability, number of network interfaces, storage capacity, and number of email domains supported. The following table details the characteristics for each supported appliance model.

Table 1 – Appliance Model Characteristics

Model	Email Msgs/Hr	Network Interfaces	Storage Capacity	Email Domains
200D	200K	4 x 10/100/1000 RJ45	1 x 1TB	50
1000D	1.7M	6 x 10/100/1000 RJ45 and 2 x SFP	2 x 2TB	3,000
3000D	2.3M	4 x 10/100/1000 RJ45 and 2 x GbE SFP	2 x 2TB (6 x 2TB Optional)	5,000

1.5.1.1 Guidance Documentation

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

1. FortiMail Secure Messaging Platform 5.2 Administration Guide
2. FortiMail Secure Messaging Platform 5.2 CLI Reference
3. FortiMail Secure Messaging Platform 5.0 Log Message Reference
4. FortiMail-200D QuickStart Guide
5. FortiMail-1000D QuickStart Guide
6. FortiMail-3000D QuickStart Guide
7. FIPS 140-2 and Common Criteria Compliant Operation for FortiMail 5.2.6

1.5.2 Logical Boundary

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

1.5.2.1 Security Audit

The TOE is capable of generating and securely transmitting Security Audit logs to a remote, trusted server for further processing and review. The TOE will generate auditable event as specified in the NDPP which may help indicate a number of potential security concerns including resonance, password guessing and tampering with the trusted paths and channels. For all applicable auditable events the TOE will associate a user (either IP address or with administrative credentials) to the session and use this identifier for all logging to the audit server.

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

1.5.2.2 Cryptographic Support

The TOE is capable of generating cryptographic keys using a NIST SP 800-90A compliant random bit generator. These keys are created, managed and destroyed to provide cryptographic services to the network. The TOE is also capable of importing cryptographic keys and certificates from outside the TOE boundary. Cryptographic keys as well as other CSPs[1] are zeroized when no longer required and the TOE offers a function to zeroize this data on demand.

The TOE is designed such that the cryptographic keys and other CSPs are not exposed through the various interfaces made available to the TOE administrator(s). Administrative passwords are encoded by the TOE via a one-way function to obscure the password credentials. Certificates are not viewable from any interface and may only be imported to the TOE through HTTPS which is a cryptographically-protected trusted and validated channel.

The TOE implements HTTPS functionality for administrator access. Additionally the TOE protects audit record communications with FortiAnalyzer via TLS.

1.5.2.3 User Data Protection

The TOE ensures that all information is zeroized on allocation of memory to ensure that all memory is cleared of residual information prior to being written to.

1.5.2.4 Identification and Authentication

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

Authentication data is protected via an encrypted trusted path. Any individual attempting to log on for an interactive session will be shown a warning message that they must accept prior to being presented with a prompt to attempt their authentication.

Credential validation is performed locally.

1.5.2.5 Security Management

The TOE provides remote and local administrative interfaces that permit the administrative roles to configure and manage the TOE both locally and remotely. An administrator account is associated with an access profile, which determines the permissions of the individual administrator. Additionally, each system comes with a default administrator account with all permissions, which may not be deleted. The term ‘authorized administrator’ is used throughout this ST to describe an administrator given the appropriate permission to perform tasks as required.

These tasks include configuring appropriate cryptographic protocols, the capacity to query the version information and the ability to update the TOE to a new version.

1.5.2.6 Protection of the TSF

Inter-TSF communications are protected to ensure availability, confidentiality and detection of modification. This is accomplished through the usage of strong cryptographic communications for

[1] Critical Security Parameters

remote administrators and audit record destinations. By default detection of modification and audit logging is enabled on TLS connections.

The TOE prevents the reading of all administrator passwords, pre-shared keys, symmetric keys and private keys through encoding them with a one-way function prior to storing them into the TOE configuration file. These keys are viewable only as this encoded value and the value will be shown when a full configuration is shown or backed up by the administrator.

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

The TOE maintains its own timestamp which is free from interference for the purposes of generating its audit logs and other time-sensitive operations on the TOE such as cryptographic key regeneration intervals and authentication timeouts and lockouts. NTP is also supported to synchronize the operating system time with an external time server.

The TOE implements a number of self-tests on start-up to ensure the correct operation and configuration of the TOE. These include but are not limited to hardware and entropy source self-tests, checksums of the binaries and operation of the cryptographic module.

1.5.2.7 TOE Access

The TOE is capable of terminating both local and remote administrative sessions upon detection of administrator inactivity. The TOE is also capable of terminating a remote session upon request from a remote administrator.

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

1.5.2.8 Trusted Path/Channels

Cryptographically-protected trusted communications channels are required for communication with the audit server. In the evaluated configuration the TOE secures its audit server communications via TLS. The TOE initiates these cryptographically protected channels.

The TOE will ensure that HTTPS (using TLS) is used for a trusted path between the TOE and the trusted remote administrator. This path will be used for both the initial administrator authentication and all remote administration requests

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

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

- Management applications including
    - Local Console Software (Serial Console client)

    - Web Browser

- Logging Server
    - FortiAnalyzer appliance

1.5.4 Evaluated Configuration

The following configuration constraints must be adhered to:

- The Fortinet Entropy Token option for FortiMail must be included. The system is configured to require the Fortinet Entropy Token to be enabled as the entropy-token.
- The system is configured to be in "FIPS-CC" mode in order to cause the cryptographic functionality of the TOE to operate in the manner stated in this document.
- Only AES with 128-bit or 256-bit keys is used for encryption/decryption.
- Cryptographic algorithms that are not CAVP-validated are not used.
- All administrator accounts are assigned to the System domain.
- Administrator authentication is performed locally by the TOE.
- Administrators always enable password policy enforcement, and it is applied to Administrator logins.
- The login disclaimer message is applied to Admin logins.
- HTTP, Telnet and SSH services are disabled; HTTPS services may be enabled on a per-network interface basis.
- Local and/or remote logging must be enabled; have a log level of Information; and the following event types must be selected for logging: When configuration has changed, Admin login/logout event, System activity event, and Update.
- The configuration for each remote logging destination must specify a TLS security profile with TLS level of Encrypt, checking encryption strength enabled, and minimum encryption strength of 128.
- The default disk partition structure is used (commands to change the disk partition structure are not used).

1.5.5 Functionality Excluded from the Evaluated Configuration

In addition to the 200D, 1000D and 3000D, FortiMail is also supported on the following appliance models: 200E, 400E, 2000B, 3000C and 5002B. These additional models were not evaluated.

In addition to local administrator authentication, the TOE also supports integration with LDAP authentication servers.

The email processing functionality of FortiMail was not evaluated. This includes Identity-Based Encryption (IBE), S/MIME, and TLS email encryption options.

FortiMail may be operated in high availability configurations with redundant appliances. This functionality was not evaluated.

2 CONFORMANCE CLAIMS

2.1 Common Criteria Conformance Claim

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

2.2 Protection Profile Conformance Claim

The Security Target claims exact conformance to the:

- Network Devices Protection Profile (NDPP) v1.1, June 8, 2012, as updated by NDPP Errata #3 (3 November 2014), including the following optional requirements [TLS and TLS/HTTPS].

3 Security Problem Definition

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

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

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

3.1 Threats

This section identifies the threats to the assets against which protection is required by the TOE or by the security environment.

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

Table 2 – Threats

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

3.2 Organizational Security Policies

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

Table 3 – Organizational Security Policies

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

3.3 Assumptions

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

Table 4 – Assumptions

Assumption	Description
A.NO_GENERAL_PURPOSE	It is assumed that there are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
A.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is assumed to be provided by the environment.
A.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.

4 Security Objectives

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

4.1 Security Objectives for the TOE

The IT security objectives for the TOE are as follows:

Table 5 – TOE Security Objectives

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

4.2 Security Objectives for the Operational Environment

The security objectives for the operational environment are addressed below:

Table 6 – Operational Environment Security Objectives

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

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

5 EXTENDED SECURITY REQUIREMENT COMPONENTS DEFINITION

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

5.1 Extended TOE Security Functional Requirement Components

This section specifies the extended SFRs for the TOE.

5.1.1 FAU_STG_EXT.1 External Audit Trail Storage

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

Management: FAU_STG_EXT.1

There are no management activities foreseen.

Audit: FAU_STG_EXT.1

There are no auditable events foreseen.

FAU_STG_EXT.1 External Audit Trail Storage

Hierarchical to: No other components

Dependencies: FAU_GEN.1 Audit data generation
FTP_ITC.1 Inter-TSF trusted channel

FAU_STG_EXT.1.1 The TSF shall be able to [selection: transmit the generated audit data to an external IT entity, receive and store audit data from an external IT entity] using a trusted channel implementing the [selection: IPsec, TLS, TLS/HTTPS] protocol.

5.1.2 FCS_CKM_EXT.4 Cryptographic Key Zeroization

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

Management: FCS_CKM_EXT.4

There are no management activities foreseen.

Audit: FCS_CKM_EXT.4

There are no auditable events foreseen.

FCS_CKM_EXT.4 Cryptographic Key Zeroization

Hierarchical to: FCS_CKM.4

Dependencies: FDP_ITC.1 Import of user data without security attributes, or
FDP_ITC.2 Import of user data with security attributes, or
FCS_CKM.1 Cryptographic key generation

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

5.1.3 FCS_HTTPS_EXT.1 Extended: HTTPS

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

Management: FCS_HTTPS_EXT.1

There are no management activities foreseen.

Audit: FCS_HTTPS_EXT.1

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

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

FCS_HTTPS_EXT.1 Extended: HTTPS

Hierarchical to: No other components

Dependencies: FCS_TLS_EXT.1 Extended: TLS

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

FCS_HTTPS_EXT.1.2 The TSF shall implement the HTTPS protocol using TLS as specified in FCS_TLS_EXT.1.

5.1.4 FCS_RBG_EXT.1 Extended: Random Bit Generation

FCS_RBG_EXT.1 Extended: Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source. It is modeled after FCS_COP.1, but belongs to a new family defined for the FCS Class.

Management: FCS_RBG_EXT.1

There are no management activities foreseen.

Audit: FCS_RBG_EXT.1

There are no auditable events foreseen.

FCS_RBG_EXT.1 Extended: Random Bit Generation

Hierarchical to: No other components

Dependencies: None

FCS_RBG_EXT.1.1 The TSF shall perform all random bit generation (RBG) services in accordance with [selection: choose one of: NIST Special Publication 800-90 using [selection: Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)]; FIPS Pub 140-2 Annex C: X9.31 Appendix 2.4 using AES] seeded by an entropy source that accumulated entropy from [selection, one or both of: a software-based noise source; a TSF-hardware-based noise source].

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

5.1.5 FCS_TLS_EXT.1 Extended: TLS

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

Management: FCS_TLS_EXT.1

There are no management activities foreseen.

Audit: FCS_TLS_EXT.1

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

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

FCS_TLS_EXT.1 Extended: TLS

Hierarchical to: No other components

Dependencies: FCS_COP.1(1) Cryptographic operation (for data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (for cryptographic signature)
FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)
FCS_COP.1(4) Cryptographic operation (for keyed-hash message authentication)
FCS_RBG_EXT.1 Extended: Cryptographic Operation (Random Bit Generation)
FCS_CKM.1 Cryptographic Key Generation
FCS_CKM_EXT.4 Cryptographic Key Zeroization

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols [selection: TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA
[selection:
None
TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384
].

5.1.6 FIA_PMG_EXT.1 Password Management

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

Management: FIA_PMG_EXT.1

There are no management activities foreseen.

Audit: FIA_PMG_EXT.1

There are no auditable events foreseen.

FIA_PMG_EXT.1 Password Management

Hierarchical to: No other components

Dependencies: None

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

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

5.1.7 FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

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

Management: FIA_UAU_EXT.2

There are no management activities foreseen.

Audit: FIA_UAU_EXT.2

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

a) All use of the authentication mechanisms.

FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

Hierarchical to: FIA_UAU.5

Dependencies: FIA_UID.1

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, [selection: [assignment: *other authentication mechanism(s)*], none] to perform user authentication.

5.1.8 FIA_UIA_EXT.1 Extended: Password-based Authentication and Identification Mechanism

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

Management: FIA_UIA_EXT.1

There are no management activities foreseen.

Audit: FIA_UIA_EXT.1

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

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

FIA_UIA_EXT.1 Extended: Password-based Authentication and Identification Mechanism

Hierarchical to: FIA_UID.1 Timing of identification
FIA_UAU.1 Timing of Authentication

Dependencies: FTA_TAB.1

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

- ○ Display the warning banner in accordance with FTA_TAB.1;
- ○ [selection: <u>no other actions, [assignment: *list of services, actions performed by the TSF in response to non-TOE requests.*</u>]]

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

5.1.9 FPT_APW_EXT.1 Extended: Protection of Administrator Passwords

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

Management: FPT_APW_EXT.1

There are no management activities foreseen.

Audit: FPT_APW_EXT.1

There are no audit activities foreseen.

FPT_APW_EXT.1 Extended: Protection of Administrator Passwords

Hierarchical to: No other components

Dependencies: None

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

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

5.1.10 FPT_SKP_EXT.1 Extended: Protection of TSF data (for reading of all symmetric keys)

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

Management: FPT_SKP_EXT.1

There are no management activities foreseen.

Audit: FPT_SKP_EXT.1

There are no audit activities foreseen.

FPT_SKP_EXT.1 Extended: Protection of TSF data (for reading of all symmetric keys)

Hierarchical to:	No other components
Dependencies:	None
FPT_SKP_EXT.1.1	The TSF shall prevent reading of all pre-shared keys, symmetric keys, and private keys.

5.1.11 FPT_TST_EXT.1 Extended: TSF testing

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

Management: FPT_TST_EXT.1

There are no management activities foreseen.

Audit: FPT_TST_EXT.1

There are no audit activities foreseen.

FPT_TST_EXT.1 TSF testing

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

5.1.12 FPT_TUD_EXT.1 Extended: Management of TSF Data

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

Management: FPT_ TUD_EXT.1

There are no management activities foreseen.

Audit: FPT_ TUD_EXT.1

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

a) Initiation of update.

FPT_ TUD_EXT.1 Extended: Trusted Update

Hierarchical to: No other components

Dependencies: [selection: FCS_COP.1(2) Cryptographic operation (for cryptographic signature), FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)]

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

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

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

5.1.13 FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking

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

Management: FTA_SSL_EXT.1

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

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

Audit: FTA_SSL_EXT.1

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

a) Any attempts at unlocking an interactive session.

FTA_SSL_EXT.1 Extended: TSF-initiated Session Locking

Hierarchical to: No other components

Dependencies: FIA_UIA_EXT.1 Password-based Authentication and Identification Mechanism

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

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

after a Security Administrator-specified time period of inactivity.

5.2 Extended TOE Security Assurance Requirement Components

There are no extended TOE Security Assurance Requirement Components.

6 Security Requirements

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

6.1 Security Functional Requirements

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

Table 7 – TOE Security Functional Requirements

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

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

6.1.1 Security Audit (FAU)

6.1.1.1 FAU_GEN.1 Audit Data Generation

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

a) Start-up and shutdown of the audit functions;
b) All auditable events for the [not specified] level of audit;
c) [*All administrative actions*]; and
d) [*Specifically defined auditable events listed in* Table *8*]

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

a) Date and time of the event, type of event, subject identity, and the outcome (success or failure) of the event; and
b) For each audit event type, based on the auditable event definitions of the functional components included in the PP/ST, [*the information detailed in Table 8*].

Table 8 – Auditable Events

Requirements	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None	None
FAU_GEN.2	None	None
FAU_STG_EXT.1	None	None
FCS_CKM.1	None	None
FCS_CKM_EXT.4	None	None
FCS_COP.1(1)	None	None
FCS_COP.1(2)	None	None
FCS_COP.1(3)	None	None
FCS_COP.1(4)	None	None
FCS_HTTPS_EXT.1	Failure to establish a HTTPS Session Establishment/Termination of a HTTPS session	Reason for failure Non-TOE endpoint of connection (IP address) for both successes and failures
FCS_RBG_EXT.1	None	None

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

6.1.1.2 FAU_GEN.2 User Identity Association

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

6.1.1.3 FAU_STG_EXT.1 External Audit Trail Storage

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

6.1.2 Cryptographic Support (FCS)

6.1.2.1 FCS_CKM.1 Cryptographic Key Generation

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

[

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

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

6.1.2.2 FCS_CKM_EXT.4 Cryptographic Key Zeroization

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

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

FCS_COP.1.1(1) **Refinement:** The TSF shall perform [*encryption and decryption*] in accordance with a specified cryptographic algorithm [*AES operating in [**CBC**[2]]*] and cryptographic key sizes [*128-bits and 256-bits*] that meets the following: [

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

Application Note: CAVP AES Certificate #3500

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

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

that meets the following:

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

Application Note: CAVP RSA Certificate #1801

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

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

Application Note: CAVP SHA Certificate #2892

[2] CBC: Cipher Block Chaining

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

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

Application Note: CAVP HMAC Certificate #2239

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

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

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

Application Note: CAVP DRBG Certificate #873

6.1.2.8 FCS_HTTPS_EXT.1 Extended: HTTPS

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

FCS_HTTPS_EXT.1.2 The TSF shall implement the HTTPS protocol using TLS as specified in FCS_TLS_EXT.1.

Application Note: CAVP CVL Certificate #574

6.1.2.9 FCS_TLS_EXT.1 Extended: TLS

FCS_TLS_EXT.1.1 The TSF shall implement one or more of the following protocols [TLS 1.0 (RFC 2246), TLS 1.1 (RFC 4346), TLS 1.2 (RFC 5246)] supporting the following ciphersuites:

TLS_RSA_WITH_AES_128_CBC_SHA
[TLS_RSA_WITH_AES_256_CBC_SHA
TLS_DHE_RSA_WITH_AES_128_CBC_SHA
TLS_DHE_RSA_WITH_AES_256_CBC_SHA
TLS_RSA_WITH_AES_128_CBC_SHA256
TLS_RSA_WITH_AES_256_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_128_CBC_ SHA256
TLS_DHE_RSA_WITH_AES_256_CBC_ SHA256].

Application Note: CAVP CVL Certificate #574

6.1.3 User Data Protection (FDP)

6.1.3.1 FDP_RIP.2 Full Residual Information Protection

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

6.1.4 Identification and Authentication (FIA)

6.1.4.1 FIA_PMG_EXT.1 Password Management

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

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

6.1.4.2 FIA_UIA_EXT.1 User Identification and Authentication

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

- Display the warning banner in accordance with FTA_TAB.1;
- [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.

6.1.4.3 FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

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

6.1.4.4 FIA_UAU.7 Protected Authentication Feedback

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

6.1.5 Security Management (FMT)

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

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

Application Note: FortiMail provides a single "administrator" role that addresses all of the administrator roles in the PP.

6.1.5.2 FMT_SMF.1 Specification of Management Functions

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

- *Ability to administer the TOE locally and remotely;*
- *Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;*
- *[Ability to configure the cryptographic functionality]*

6.1.5.3 FMT_SMR.2 Restrictions on Security Roles

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

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

FMT_SMR.2.3 The TSF shall ensure that the conditions [

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

are satisfied.

Application Note: FortiMail provides a single "administrator" role that addresses all of the administrator roles in the PP.

6.1.6 Protection of the TSF (FPT)

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

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

6.1.6.2 FPT_APW_EXT.1 Extended: Protection of Administrator Passwords

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

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

6.1.6.3 FPT_STM.1 Reliable Time Stamps

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

6.1.6.4 FPT_TUD_EXT.1 Extended: Trusted Update

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

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

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

6.1.6.5 FPT_TST_EXT.1: TSF Testing

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

6.1.7 TOE Access (FTA)

6.1.7.1 FTA_SSL_EXT.1 TSF-initiated Session Locking

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

6.1.7.2 TSF-initiated Termination

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

6.1.7.3 FTA_SSL.4 User-initiated Termination

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

6.1.7.4 FTA_TAB.1 Default TOE Access Banners

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

6.1.8 Trusted Path/Channels (FTP)

6.1.8.1 FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 **Refinement**: The TSF shall **use [TLS]** to provide a **trusted** communication channel between itself and **authorized IT entities supporting the following capabilities: [audit server]** 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 [logging of audit messages, validation of administrator credentials].

Application Note: The clause "or the authorized IT entities" is included for PP conformance. All trusted channel communication is initiated by the TSF.

6.1.8.2 FTP_TRP.1 Trusted Path

FTP_TRP.1.1 **Refinement**: The TSF shall **use [TLS/HTTPS]** provide a **trusted** communication path between itself and **remote administrators** that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from [disclosure and **detection of modification of the communicated data**].

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

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

6.2 Security Assurance Requirements

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

Table 9 – Security Assurance Requirements

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

7 TOE SUMMARY SPECIFICATION

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

7.1 Security Audit

For all administrative actions, including management of the TOE, authentication is required before any actions can occur on the TOE. When an action identified in Table 8 is triggered the TOE will write the event including the administrative username of the user triggering the event to the audit log.

In the evaluated configuration the event log is always considered to be on and logging once the TOE is fully initialized and services are available in normal operation. The TOE logs the startup and shutdown of the TOE, and this can be considered to be equivalent to the startup and shutdown of the audit system.

The TOE is capable of simultaneously logging the audit messages both locally and remotely, and has configurable actions when the local audit logs are filled. By default the TOE will log locally and will block further traffic from occurring should the local storage become exhausted. Guidance is provided to the administrator to modify this behavior to overwrite the oldest audit logs upon hitting a threshold of memory capacity. The most recent audit records are stored locally. Only authorized administrators may view these records, and no capability to modify the records is provided. 80% of the appliance disk capacity (e.g. 80% of 1TB for the 400C) is reserved for local audit log storage. The disk capacity for each appliance model is provided in Table 1.

The TOE has configurable options for the remote storage of the audit events. These events are sent in real-time to one or more configured audit servers. In the evaluated configuration these audit servers can be FortiAnalyzer analytics suite secured through the usage of TLS. These audit events are transmitted as they are generated; a cache accommodating 32K audit records is maintained to address temporary outages in communication with remote audit servers. If the cache is exhausted the oldest record is discarded in order to make room for new records.

The TOE is capable of logging messages to the audit log for interactions which occur via HTTPS. These events include attempts to establish or terminate sessions and errors detected during decryption or validation of data.

7.2 Cryptographic Support

The TOE uses CAVP-validated cryptography that has been implemented in the Fortinet FortiMail RNG Cryptographic Library Version 5.2 (RNG/RBG only) and the Fortinet FortiMail SSL Cryptographic Library Version 5.2 (all other algorithms). These libraries are incorporated in the FortiMail-200D, FortiMail-1000D and FortiMail-3000D.

The TOE only stores keys in memory, either in RAM or Flash memory. Keys are destroyed by overwriting the key storage once with a fixed pattern.

The TOE is capable of generating full entropy by using a full entropy seed provided to the random bit generator from Fortinet Entropy Token in order to provide cryptographic services to the TOE. The TOE is also capable of importing cryptographic keys and certificates from outside the TOE boundary. These keys are zeroized when no longer required and the TOE offers a function to zeroize these keys on demand.

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

Table 10 – Cryptographic Algorithms

Cryptographic Operations	Cryptographic Algorithm	Key Sizes (bits)	Message Digest Size	FIPS Standard	Certificate #
Symmetric Encryption and Decryption	AES operating in CBC	128, 256	N/A	FIPS PUB 197 (AES) NIST SP800-38A	CAVP Certificate # (3500)
Cryptographic Hashing	SHA-1, SHA-256	160, 256	160, 256	FIPS Pub 198-1 (HMAC) FIPS Pub 180-3 (SHS)	CAVP Certificate # (2892)
Keyed-Hash message authentication (HMAC)	SHA-1, SHA-256	160, 256	160, 256	FIPS Pub 180-3 (SHS)	CAVP Certificate # (2239)
Signature Verification	rDSA	2048, 3072	N/A	FIPS PUB 186-3 (DSS) FIPS PUB 186-2 (DSS)	CAVP Certificate # (1801)
Signature Generation	rDSA	2048, 3072	N/A	FIPS PUB 186-3 (DSS) FIPS PUB 186-2 (DSS)	CAVP Certificate # (1801)
DRBG	CTR_DRBG (AES)	256	N/A	NIST SP800-90	CAVP Certificate # (873)

As part of the Fortinet FortiMail SSL Cryptographic Library testing, the Diffie-Hellman Ephemeral (DHE) implementation (via the CAVS tool) was validated for the correctness of the implementation as both an initiator and responder. For additional details on Diffie-Hellman or any other cryptographic operations, please refer to the appropriate certificate as stated in the table above.

7.2.1 Entropy Source and Random Bit Generation

The TOE implements an entropy collection system from a hardware based Fortinet Entropy Token noise source which is derived from wide-band RF white noise which is then pooled and conditioned prior to being used. This noise source provides full entropy to the random number generation up to 256 bits.

The Fortinet FortiMail RNG Cryptographic Library Version 5.2 contains a CTR_DRBG implemented per NIST SP 800-90A (CAVP DRBG Certificate #873) and is seeded with a hardware entropy source (Fortinet Entropy Token). Entropy from the noise source are extracted 5120 bits at a time, conditioned and 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.

7.2.2 Cryptographically Trusted Channels

Trusted channels are used to protect all data exchanges with remote administrators (including credentials). Remote administration sessions apply to the Network Web-Based GUI.

7.2.3 HTTPS

The Network Web-Based GUI uses the HTTPS protocol for secure administrator communications. With respect to the TOE implementation of HTTPS, TLS version 1.0, 1.1 or 1.2 is used to encrypt and authenticate administration sessions between the remote browser and TOE. The TOE supports the following ciphersuites:

- TLS_RSA_WITH_AES_128_CBC_SHA,
- TLS_RSA_WITH_AES_256_CBC_SHA,
- TLS_RSA_WITH_AES_128_CBC_SHA256,
- TLS_RSA_WITH_AES_256_CBC_SHA256,
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA,
- TLS_DHE_RSA_WITH_AES_128_CBC_SHA256 ,
- TLS_DHE_RSA_WITH_AES_256_CBC_SHA256.

TLS 1.0, 1.1 or 1.2 is also used for the purposes of protecting the audit logs while in transit to the FortiAnalyzer audit servers.

7.2.4 TLS

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

- Server sends 2048-bit RSA public certificate
- Server generates, signs (RSA PKCS#1) and sends DH parameters and DH public value
- Client generates and sends DH public value. The keying material is then used to encrypt/decrypt (AES128 and AES256) and authenticate (HMAC-SHA1 or HMAC-SHA256) the data exchange.

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

The trusted channels protect communication between the TOE and remote audit servers. These paths are logically distinct from other communication channels and provide assured identification of the end points and protection of the data from modification and disclosure.

7.2.5 Cryptographic Self Tests and TOE Update Integrity

The TSF provides a cryptographic function that an administrator may use to verify the integrity of the TSF executable code. During a normal boot-up sequence the TOE administrator can see on the local console the following types of tests:

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

Indication of successful tests would appear as follows:

Running <test>... passed

Completion of all self-tests is indicated by:

Self-tests passed

The TOE is capable of running these tests at the request of an administrator, and periodically at an administrator-specified interval not less than once a day to demonstrate the correct operation of the cryptographic components of the TSF. The TOE will enter into an Error Mode when failure of a self-test (integrity verification self-test, or cryptographic self-test) is detected. This mode allows the TOE to enter into a secure state. These self-tests are executed on initial start-up or at the request of an administrator. Upon successful completion of these tests an audit log will be generated by the TOE and sent to the remote audit server.

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

7.2.6 Conformance to NIST SP800-56

While the TOE fulfills all of the NIST SP 800-56B requirements without extensions, the following table specifically identifies the “should”, “should not”, and “shall not” conditions from the publication along with an indication of whether the TOE conforms to those conditions with deviations rationalized.

Table 11 - NIST SP800-56B Conformance

Section #	“should”, “should not”, or “shall not”	Implemented accordingly?	Rationale for deviation
5.6	should	Yes	
5.8	shall not	Yes	
5.9	shall not (first occurrence)	Yes	
5.9	shall not (second occurrence)	Yes	
6.1	should not	Yes	
6.1	should (first occurrence)	Yes	

Section #	"should", "should not", or "shall not"	Implemented accordingly?	Rationale for deviation
6.1	should (second occurrence)	Yes	
6.1	should (third occurrence)	Yes	
6.1	should (fourth occurrence)	Yes	
6.1	shall not (first occurrence)	Yes	
6.1	shall not (second occurrence)	Yes	
6.2.3	should	Yes	
6.5.1	should	Yes	
6.5.2	should	Yes	
6.5.2.1	should	Yes	
6.6	shall not	Yes	
7.1.2	should	Yes	
7.2.1.3	should	Yes	
7.2.1.3	should not	Yes	
7.2.2.3	should (first occurrence)	Yes	
7.2.2.3	should (second occurrence)	Yes	
7.2.2.3	should (third occurrence)	Yes	
7.2.2.3	should (fourth occurrence)	Yes	
7.2.2.3	should not	Yes	
7.2.2.3	shall not	Yes	
7.2.3.3	should (first occurrence)	Yes	
7.2.3.3	should (second occurrence)	Yes	
7.2.3.3	should (third occurrence)	Yes	
7.2.3.3	should (fourth occurrence)	Yes	
7.2.3.3	should (fifth occurrence)	Yes	
7.2.3.3	should not	Yes	
8	should	Yes	
8.3.2	should not	Yes	

7.2.7 Key and CSP storage and zeroization

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

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

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

These CSPs are cleared when the appliance power cycles or reboots. Ephemeral keys are overwritten with a fixed pattern 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.

The following table provides details of the storage locations and zeroization methods used by the TOE for keys and CSPs.

Table 12 – Key/CSP Storage and Zeroization

Key or CSP	Storage	Zeroization Method
NDRNG input string	RAM	Power cycle or reboot
DBRG seed	RAM	Power cycle or reboot
DRBG output	RAM	Power cycle or reboot
DBRG v and key values	RAM	Power cycle or reboot
Diffie-Hellman Key	RAM	Power cycle or reboot
Firmware Update Key	Flash storage	Format flash storage
Firmware Integrity Key	Flash storage	Format flash storage
HTTPS/SSL Server/Host Key	Flash storage	Format flash storage
HTTPS/TLS Session Authentication Key	RAM	Power cycle or reboot; session terminated
HTTPS/TLS Session Encryption Key	RAM	Power cycle or reboot; session terminated
Configuration Integrity Key	Flash storage	Format flash storage
Configuration Encryption Key	RAM	Power cycle or reboot
Configuration Backup Key	Flash storage	Format flash storage
Operator Password	Flash storage	Factory reset
User Password	Flash storage	Factory reset

7.3 User Data Protection

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

7.4 Identification and Authentication

The TOE supports a variety of methods of Identification and Authentication to both local and external sources. Regardless of the method of administration that is chosen by the administrator no administrative action is possible prior to authentication.

The TOE uses a local password database for all of its locally-defined credentials. Passwords can be created through the usage of mixed case characters, digits and the special characters "!", "@", "#", "$", "%", "^", "&", "*", "(" and ")".

7.4.1 Web/HTTPS

By default the web/HTTPS interface is enabled on the first network port. The TOE may also be configured to allow or disallow access to this TSFI on a per-network port basis in either the CLI or the web UI. The HTTPS web interface is accessed by going to the TOE IP on port 443. Once connected to the port and the HTTPS session is established the TOE provides a warning banner according to FTA_TAB.1 which the administrator must accept prior to proceeding. Following this banner the user will then accept the warning and be presented with a username and login screen. During the credential entry the user's password is entered in a "password" input box, and only dots are echoed. When complete, the credentials are sent to the TOE (protected by TLS). If the supplied username is not a configured administrator account, the login fails. If the supplied username is valid, the configured authentication mechanism configured for that account is retrieved. For local authentication, the local credential store is consulted and if there is a match access is granted to the TOE (the initial GUI is displayed to the user). A failed login attempt will be met with the '' Authentication failure. Please try again..." error message.

7.4.2 Local Console

The local console is only accessible through the use of the dedicated management port present on the TOE and requires that the management station be appropriately configured. Depending on the hardware model this could be via Serial Console or USB.

Local access is enabled and may not be disabled. First the pre-login warning banner is displayed as configured by the administrator. Next the user is prompted for their username which is echoed back to the screen. Following the identification the user is requested to put in their password which is hidden and provides no feedback indicating any progress. Once the credentials have been entered the TOE validates them. If the supplied username is not a configured administrator account, the login fails. If the supplied username is valid, the configured authentication mechanism configured for that account is retrieved. For local authentication, the local credential store is consulted and if there is a match access is granted to the TOE (the CLI prompt is displayed to the user). Following successful authentication the TOE will change the command prompt to the hostname followed by #. If the supplied credentials are not valid, the user is prompted again for their password. After 4 consecutive failed login attempts, a one minute delay is imposed by the TOE before the user is again prompted for their username.

7.5 Security Management

The security management for the TOE is implemented on a per-interface basis. Regardless of the interface no management functionality is possible prior to authentication. The TOE is capable of having custom roles defined however, only the *Authorized Administrator* role who can administer all functionality of the TOE is defined.

7.5.1 CLI (Local Console)

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

Once authenticated the CLI gives administrators full control over all aspects of the TOE including the management and setting of users and cryptographic operations.

7.5.2 Web UI

The TOE tracks administrative sessions on the WebUI through the use of cookies and a session database on the TOE. When an administrator logs onto the TOE the cookies and session database are consulted to determine if there is already an open session for this instance. In the event that there is no pre-existing session established for the management of the TOE the user is redirected to the login page. Stale administrative sessions are logged out after a period of inactivity to ensure that unattended administrator sessions can’t be hijacked.

Once authenticated the WebUI gives administrators full control over all aspects of the TOE including the management and setting of users and cryptographic operations.

7.6 Protection of the TSF

The TOE uses a number of methods to protect itself and the communications channels which it provides from potentially hostile entities. An internal clock source with battery backup in the hardware is used to initialize the time maintained by the operating system at boot. Subsequently hardware interrupts generated at fixed intervals are used to update the time accurately. NTP is also supported to synchronize the operating system time with an external time server. The time maintained by the operating system is used to:

- generate time stamps in audit records,
- keying and keylife timeouts, and
- authentication timeouts and lockouts.

7.6.1 Cryptographic Key and Password Storage

The TOE incorporates a cryptographic module that protects any keys provided to or stored within the cryptographic module. Cryptographic keys within this module are not capable of being viewed through the CLI or Web interface. The TOE does not provide any method of direct access to view or modify files over either of these interfaces.

Cryptographic keys related to the HTTPS interface are stored on the local file system of the TOE. An authorized administrator can generate a certificate signing request from the TOE and import the signed certificate back into the TOE. Once a CSP is imported into the TOE this information cannot be viewed again through any of the TSFI's. These keys can be zeroized through the methods described in section 7.2.7.

Pre-shared keys related to the administrator passwords and other credentials for the secure operation of the TOE are stored in the TOE's configuration file. Authorized administrators are allowed to enter this information through one of the protected communications paths such as the CLI or HTTPS GUI. Once the password is entered the TOE hashes the password to the configuration file permanently obscuring the contents. This configuration file with the obscured password hashes is available through the local console and HTTPS GUI by viewing a full configuration or backup of the configuration.

A detailed description of the storage locations and zeroization methods of keys and CSP is provided in Table 12.

7.6.2 Product Updates

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 via a trusted path such as the HTTPS web interface.

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

1. If a signature is not present, abort the upgrade
2. Extract the public key and signature from the firmware
3. Validate that the public key is the same as is stored on the TOE. If the public keys do not match abort the upgrade.
4. 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.

7.6.3 Self-Tests

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

- CPU and Memory BIOS self-tests – CPU and memory are initialized by exercising a set of known answer tests and the BIOS is compared against a known checksum of the image. The memory is zeroized and then has a random pattern written and read from the memory.
- Boot loader image verification – the boot loader will compare the image of the TOE to a known checksum of the image prior to booting.
- Noise source tests – the noise source is started and pattern analysis is done on the output to ensure that the source is not in a cryptographically weak state. These include both the repetition and adaptive proportion tests.

- Known Answer Tests (KAT) – comparison of a number of cryptographic functions against an expected set of values

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

- KAT (as described above)
- Noise source (as described above)

The TOE also performs the following ongoing self-tests

- Noise source pattern analysis

If any test fails, error messages (“Self-tests failed” and “Entering error mode....”) are displayed on the console and the system halts.

This collection of tests verifies that the hardware is operating as intended and that the software being executed has not been modified. Since the software is known to be a tested and approved version (whether obtained via product delivery or trusted update), and the hardware is known to be working properly, assurance is provided that the TSF is working properly.

7.7 TOE Access

The TOE has a number of methods to restrict access to only those administrators who are authorized to administer the TOE. The first is a login warning prior to allowing a user to log in stating a message configured by an authorized administrator. This message is presented on the local console as well as the HTTPS connections.

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

7.8 Trusted Path/Channels

The TOE is designed for secure operation from a trusted administrator to ensure correct operation of the TOE. Additionally the TOE can secure communications to a remote audit or analysis server such as FortiAnalyzer through TLS.

Trusted paths for an administrator to communicate with the TOE are implemented through the HTTPS interface. These are the only methods of remote communication with the TOE. When a remote administrator initiates a connection via either of these interfaces the TOE will establish a cryptographically strong communication path to the workstation of the administrator which will be used for all communication between the TOE and the authorized administrator. The TOE will detect, log and reject any packets which indicate that the communications of this path have been tampered with or modified.

Trusted channels are provided for the purposes of securing the storage of audit logs. When the TOE is configured to send the logs to FortiAnalyzer, communications are secured by TLS 1.0, 1.1 or 1.2. When an auditable event which is required to be written to the remote audit server is generated the TOE connects to the audit server over TLS and writes the event log to the audit server. No persistent connection is maintained with the audit server.

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

8 RATIONALE

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

The dependency rationale is not stated by the NDPP, and as such is provided below.

8.1 Dependency Rationale

Table 13 –Functional Requirements Dependencies

SFR	Dependencies	Dependency Met	Rationale
FAU_GEN.1	FPT_STM.1	Yes	
FAU_GEN.2	FAU_GEN.1	Yes	
	FIA_UID.1	Yes	FIA_UIA_EXT.1 is hierarchical to FIA_UID.1, thus the dependency is satisfied
FAU_STG_EXT.1	FAU_GEN.1	Yes	
	FTP_ITC.1	Yes	
FCS_CKM.1	FCS_CKM.2 or FCS_COP.1	Yes	The TOE implements FCS_COP.1
	FCS_CKM.4	Yes	FCS_CKM_EXT.4 is hierarchical to FCS_CKM.4, thus the dependency is satisfied
FCS_CKM_EXT.4	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	Yes	The TOE implements FCS_CKM.1
FCS_COP.1(1)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	Yes	The TOE implements FCS_CKM.1
	FCS_CKM.4	Yes	FCS_CKM_EXT.4 is hierarchical to FCS_CKM.4, thus the dependency is satisfied
FCS_COP.1(2)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	Yes	The TOE implements FCS_CKM.1
	FCS_CKM.4	Yes	FCS_CKM_EXT.4 is hierarchical to FCS_CKM.4, thus the dependency is satisfied
FCS_COP.1(3)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	Yes	The TOE implements FCS_CKM.1
	FCS_CKM.4	Yes	FCS_CKM_EXT.4 is hierarchical to FCS_CKM.4, thus the dependency is satisfied

SFR	Dependencies	Dependency Met	Rationale
FCS_COP.1(4)	FDP_ITC.1 or FDP_ITC.2 or FCS_CKM.1	Yes	The TOE implements FCS_CKM.1
	FCS_CKM.4	Yes	FCS_CKM_EXT.4 is hierarchical to FCS_CKM.4, thus the dependency is satisfied
FCS_HTTPS_EXT.1	FCS_TLS_EXT.1	Yes	
FCS_RBG_EXT.1	None	Yes	
FCS_TLS_EXT.1	FCS_COP.1(1)	Yes	
	FCS_COP.1(2)	Yes	
	FCS_COP.1(3)	Yes	
	FCS_COP.1(4)	Yes	
	FCS_RBG_EXT.1	Yes	
	FCS_CKM.1	Yes	
	FCS_CKM_EXT.4	Yes	
FDP_RIP.2	None	Yes	
FIA_PMG_EXT.1	None	Yes	
FIA_UAU.7	FIA_UAU.1	Yes	FIA_UIA_EXT.1 is hierarchical to FIA_UAU.1, thus the dependency is satisfied
FIA_UAU_EXT.2	None	Yes	
FIA_UIA_EXT.1	FTA_TAB.1	Yes	
FMT_MTD.1	FMT_SMR.1	Yes	
	FMT_SMF.1	Yes	
FMT_SMF.1	None	Yes	
FMT_SMR.2	FIA_UID.1	Yes	FIA_UIA_EXT.1 is hierarchical to FIA_UID.1, thus the dependency is satisfied
FPT_APW_EXT.1	None	Yes	
FPT_SKP_EXT.1	None	Yes	
FPT_STM.1	None	Yes	
FPT_TST_EXT.1	None	Yes	
FPT_TUD_EXT.1	FCS_COP.1(2) or FCS_COP.1(3)	Yes	The TOE implements signature verification via FCS_COP.1(2)

SFR	Dependencies	Dependency Met	Rationale
FTA_SSL.3	None	Yes	
FTA_SSL.4	None	Yes	
FTA_SSL_EXT.1	FIA_UIA_EXT.1	Yes	
FTA_TAB.1	None	Yes	
FTP_ITC.1	None	Yes	
FTP_TRP.1	None	Yes	

9 ACRONYMS

Table 14 – Acronyms

Acronym	Definition
AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CAVP	Cryptographic Algorithm Validation Program
CC	Common Criteria
CEM	Common Evaluation Methodology
CFB	Cipher Feedback
CSP	Critical Security Parameters
DRBG	Deterministic Random Bit Generator
ECB	Electronic Code Book
FIPS	Federal Information Processing Standard
HMAC	Keyed-Hash Message Authentication Code
NDPP	Network Device Protection Profile
OFB	Output Feedback
OSP	Organizational Security Policy
PP	Protection Profile
RBG	Random Bit Generator
SAR	Security Assurance Requirement
SFR	Security Functional Requirement
SHA	Secure Hash Algorithm
SHS	Secure Hash Standard
SSL	Secure Socket Layer
ST	Security Target
TOE	Target of Evaluation
TSF	TOE Security Functionality