Fortinet, Inc.
Common Criteria Security Target
Document Version: 1.5

Prepared By:
Acumen Security
2400 Research Blvd, Suite 395
Rockville, MD, 20850

www.acumensecurity.net

Table Of Contents

Revision History

Version	**Date**	**Description**
1.0	January 2018	Check-in package release
1.1	May 2018	Updated to address ECR comments
1.2	November 2018	Updated for check-out
1.3	December 2018	Updated based on comments
1.4	January 2019	Updated based on comments
1.5	January 2019	Updated based on CAVP comments

1 Security Target Introduction

1.1 Security Target and TOE Reference

This section provides information needed to identify and control this ST and its TOE.

Category	Identifier
ST Title	Fortinet FortiMail Appliances Security Target
ST Version	1.5
ST Author	Acumen Security, LLC.
TOE Identifier	Fortinet FortiMail Appliances running Software version 6.0.2
TOE Hardware Versions	FortiMail Appliances: FML-2000E, FML-3000E, FML-3200E
TOE Software Version	FortiMail Appliances: 6.0
TOE Developer	Fortinet, Inc.
Key Words	Network Device, Security Appliance

Table 1 TOE/ST Identification

1.2 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. 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.

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 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. 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. 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.

Fortinet Entropy Token (delivered as part of the TOE) 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.

1.2.1 TOE Product Type

Fortinet FortiMail Appliances are network devices. Each appliance runs a custom-built hardened OS with only the required services enabled.

1.2.2 Non-Evaluated Functionality

The following functionality was not evaluated as part of this NDcPPv2.0e Common Criteria evaluation,

- Antivirus
- Antispam
- Content Filtering
- Email Routing
- Email Archiving
- Mail Transfer Agent (MTA) Functionality
- SMTP/SMTPS Routing
- S/MIME/email encryption
- Identity-Based Encryption (IBE)

1.3 TOE Description

The TOE is comprised of three models of the Fortinet FortiMail Appliances as shown below.

	CPU	Storage	RAM
FML-2000E	Intel Xeon E5	2x 2TB HDD (max 8)	32GB
FML-3000E	Intel Xeon E5	2x 2TB HDD (max 12)	32GB
FML-3200E	Intel Xeon E5	2x 2TB HDD (max 12)	64GB

Table 2 FortiMail Appliances

1.4 TOE Evaluated Configuration

The TOE evaluated configuration consists of one of the appliances listed above. The TOE supports secure connectivity with several IT environment devices as shown in Table 3,

Component	Required	Usage/Purpose Description for TOE performance
Management Workstation with Web Browser	Yes	This includes any IT Environment Management workstation with a Web Browser installed that is used by the TOE administrator to support TOE administration through HTTPS protected channels.
Audit Server	Yes	The syslog audit server is used for remote storage of audit records that have been generated by and transmitted from the TOE. The syslog server must support communications using TLS 1.1 or TLS 1.2.

Table 3 IT Environment Components

The following figure provides a visual depiction of an example of a typical TOE deployment. The TOE boundary is surrounded with **red lines**.

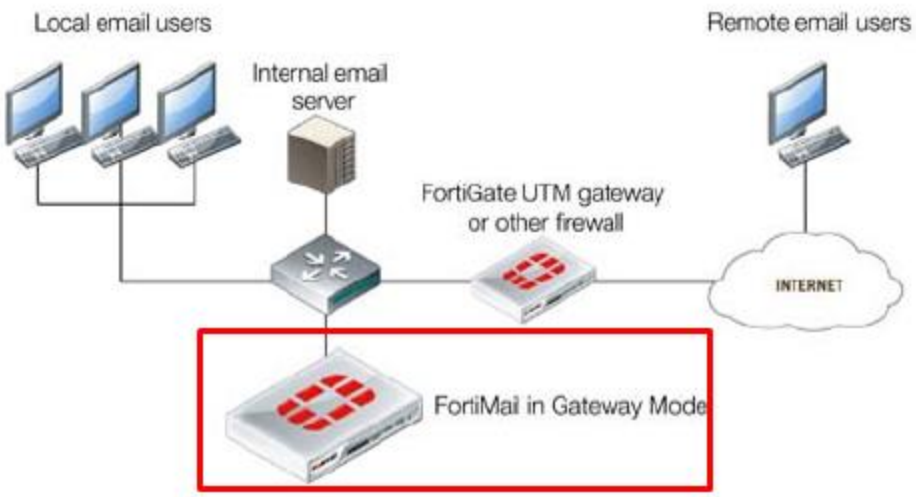

Figure 1 Physical Boundary

1.5 TOE Architecture

1.5.1 Physical Boundaries

The TOE is a hardware and software solution that is comprised of the security appliance models described above in Section 1.3. The TOE guidance documentation that is considered to be part of the TOE can be found listed in the Fortinet Fortimail v6.0 CC Guidance Documentation.

The network on which the TOE resides is considered part of the environment. The software version 6.0 is pre-installed on the TOE hardware. In addition, the software images are also downloadable from the Fortinet website. A login ID and password is required to download the software image.

1.5.2 Logical Boundaries

The TOE provides the following security functions:

- **Protected Communications.** The TOE protects the integrity and confidentiality of communications as follows:
  - TLS connectivity with the following entities:
    - Audit Server (with device level authentication)
    - Web Browser (on a management workstation)
- **Secure Administration.** The TOE enables secure local and remote management of its security functions, including:
  - Local console CLI administration
  - Remote GUI administration via HTTPS/TLS
  - Administrator authentication using a local database
  - Timed user lockout after multiple failed authentication attempts
  - Password complexity enforcement
  - Role Based Access Control – The TOE supports one default full privilege user account. However, additional accounts may be created with reduced access. For the purpose of testing, the full privileged account was used as the Security Administrator.
  - Configurable banners to be displayed at login
  - Timeouts to terminate administrative sessions after a set period of inactivity
  - Protection of secret keys and passwords

- **Trusted Update.** The TOE ensures the authenticity and integrity of software updates through digital signatures and requires administrative intervention prior to the software updates being installed.
- **Security Audit.** The TOE keeps local and remote audit records of security relevant events. The TOE internally maintains the date and time which can be set manually.
- **Self-Test.** The TOE performs a suite of self-tests to ensure the correct operation and enforcement of its security functions.
- **Cryptographic Operations.** The TOE provides cryptographic support for the services described in Table 4. The Fortinet FortiMail appliance leverages the 'Fortinet FortiMail SSL Cryptographic Library Version 6.0' and 'Fortinet FortiMail RNG Cryptographic Library 6.0' for cryptographic algorithms.

Cryptographic Method	Use within the TOE
TLS Establishment	Used to establish initial TLS session.
Signature Services	Used in TLS session establishment. Used in secure software update.
SP 800-56A Key Agreement	Used in TLS session establishment.
Key Generation	Used in TLS session establishment.
Diffie-Hellman Group 14	Used in TLS session establishment.
SP 800-90 DRBG	Used in TLS session establishment.
SHS	Used in secure software update
HMAC-SHS	Used to provide TLS traffic integrity verification
AES	Used to encrypt TLS traffic

Table 4 TOE Provided Cryptography

Additional features including SSH administration and NTP are not included in the evaluated configuration.

2 Conformance Claims

2.1 CC Conformance

This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 4, September 2012
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 4, September 2012: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 4, September 2012: Part 3 conformant

2.2 Protection Profile Conformance

This TOE is conformant to:

- collaborative Protection Profile for Network Devices (NDcPP) + Errata 20180314 version 2.0e (NDcPPv2.0e), March 14, 2018.

2.3 Scheme Interpretations

The following table identifiers the applicable technical decisions.

Identifier	Applicable	Note
0343 – NIT Technical Decision for Updating FCS_IPSEC_EXT.1.14 Tests	No	This TD addresses FCS_IPSEC_EXT.1. FCS_IPSEC_EXT.1 is not included in the ST.
0342 – NIT Technical Decision for TLS and DTLS Server Tests	Yes	
0341 – NIT Technical Decision for TLS wildcard checking	Yes	
0340 – NIT Technical Decision for Handling of the basicConstraints extension in CA and leaf certificates	Yes	
0339 – NIT Technical Decision for Making password-based authentication optional in FCS_SSHS_EXT.1.2	No	This TD addresses SSH functionality. The TOE does not support SSH.
0338 – NIT Technical Decision for Access Banner Verification	Yes	
0337 – NIT Technical Decision for Selections in FCS_SSH*_EXT.1.6	No	This TD addresses SSH functionality. The TOE does not support SSH.
0336 – NIT Technical Decision for Audit requirements for FCS_SSH*_EXT.1.8	No	This TD addresses SSH functionality. The TOE does not support SSH.
0335 – NIT Technical Decision for FCS_DTLS Mandatory Cipher Suites	Yes	While the title specifies FCS_DTLS, the update includes an update to the application note for FCS_TLS.
0334 – NIT Technical Decision for Testing SSH when password-based authentication is not supported	No	This TD addresses SSH functionality. The TOE does not support SSH.
0333 – NIT Technical Decision for Applicability of FIA_X509_EXT.3	Yes	
0324 – NIT Technical Decision for Correction of section numbers in SD Table 1	Yes	
0323 – NIT Technical Decision for DTLS server testing - Empty Certificate Authorities list	No	This TD addresses DTLS functionality. The TOE does not support DTLS.

Identifier	Applicable	Note
0322 – NIT Technical Decision for TLS server testing - Empty Certificate Authorities list	No	This TD is associated with FCS_TLSS_EXT.2. The TOE does not include FCS_TLSS_EXT.2 functionality.
0321 – Protection of NTP communications	No	The TOE does not support/include NTP.
0291 – NIT technical decision for DH14 and FCS_CKM.1	Yes	
0290 – NIT technical decision for physical interruption of trusted path/channel.	Yes	
0289 – NIT technical decision for FCS_TLSC_EXT.x.1 Test 5e	Yes	
0281 – NIT Technical Decision for Testing both thresholds for SSH rekey	No	This TD addresses SSH functionality. The TOE does not support SSH.
0259 – NIT Technical Decision for Support for X509 ssh rsa authentication IAW RFC 6187	No	This TD addresses SSH functionality. The TOE does not support SSH.
0257 – NIT Technical Decision for Updating FCS_DTLSC_EXT.x.2/FCS_TLSC_EXT.x.2 Tests 1-4	Yes	
0256 – NIT Technical Decision for Handling of TLS connections with and without mutual authentication	Yes	
0228 – NIT Technical Decision for CA certificates - basicConstraints validation	Yes	

Table 5 Technical Decisions

2.4 Conformance Rationale

This Security Target provides exact conformance to the Protection Profile(s) described in the conformance claims above. The security problem definition, security objectives and security requirements in this Security Target are all taken from the applicable Protection Profile(s) performing only operations defined there.

3 Security Problem Definition

The security problem definition has been taken from [NDcPPv2.0e] and is reproduced here for the convenience of the reader.

3.1 Threats

The threats for the Network Device are grouped according to functional areas of the device in the sections below.

3.1.1 Communications with the Network Device

A network device communicates with other network devices and other network entities. The endpoints of this communication can be geographically and logically distant and may pass through a variety of other systems. The intermediate systems may be untrusted providing an opportunity for unauthorized communication with the network device or for authorized communication to be compromised. The security functionality of the network device must be able to protect any critical network traffic (administration traffic, authentication traffic, audit traffic, etc.). The communication with the network device falls into two categories: authorized communication and unauthorized communication.

Authorized communication includes network traffic allowable by policy destined to and originating from the network device as it was designed and intended. This includes critical network traffic, such as network device administration and communication with an authentication or audit logging server, which requires a secure channel to protect the communication. The security functionality of the network device includes the capability to ensure that only authorized communications are allowed and the capability to provide a secure channel for critical network traffic. Any other communication with the network device is considered unauthorized communication. (Network traffic traversing the network device but not ultimately destined for the device, e.g. packets that are being routed, are not considered to be "communications with the network device" – cf. A.NO_THRU_TRAFFIC_PROTECTION in section 3.2.3.)

The primary threats to network device communications addressed in [the NDcPPv2.0e] focus on an external, unauthorized entity attempting to access, modify, or otherwise disclose the critical network traffic. A poor choice of cryptographic algorithms or the use of non-standardized tunnelling protocols along with weak Administrator credentials, such as an easily guessable password or use of a default password, will allow a threat agent unauthorized access to the device. Weak or no cryptography provides little to no protection of the traffic allowing a threat agent to read, manipulate and/or control the critical data with little effort. Non-standardized tunnelling protocols not only limit the interoperability of the device but lack the assurance and confidence standardization provides through peer review.

3.1.1.1 T.UNAUTHORIZED_ADMINISTRATOR_ACCESS

Threat agents may attempt to gain Administrator access to the network device by nefarious means such as masquerading as an Administrator to the device, masquerading as the device to an Administrator, replaying an administrative session (in its entirety, or selected portions), or performing man-in-the-middle attacks, which would provide access to the administrative session, or sessions between network devices. Successfully gaining Administrator access allows malicious actions that compromise the security functionality of the device and the network on which it resides.

3.1.1.2 T.WEAK_CRYPTOGRAPHY

Threat agents may exploit weak cryptographic algorithms or perform a cryptographic exhaust against the key space. Poorly chosen encryption algorithms, modes, and key sizes will allow attackers to compromise the algorithms, or brute force exhaust the key space and give them unauthorized access allowing them to read, manipulate and/or control the traffic with minimal effort.

3.1.1.3 T.UNTRUSTED_COMMUNICATION_CHANNELS

Threat agents may attempt to target network devices that do not use standardized secure tunnelling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the network device itself.

3.1.1.4 T.WEAK_AUTHENTICATION_ENDPOINTS

Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g. a shared password that is guessable or transported as plaintext. The consequences are the same as a poorly designed protocol, the attacker could masquerade as the Administrator or another device, and the attacker could insert themselves into the network stream and perform a man-in-the-middle attack. The result is the critical network traffic is exposed and there could be a loss of confidentiality and integrity, and potentially the network device itself could be compromised.

3.1.2 Valid Updates

Updating network device software and firmware is necessary to ensure that the security functionality of the network device is maintained. The source and content of an update to be applied must be validated by cryptographic means; otherwise, an invalid source can write their own firmware or software updates that circumvents the security functionality of the network device. Methods of validating the source and content of a software or firmware update by cryptographic means typically involve cryptographic signature schemes where hashes of the updates are digitally signed.

Unpatched versions of software or firmware leave the network device susceptible to threat agents attempting to circumvent the security functionality using known vulnerabilities. Nonvalidated updates or updates validated using non-secure or weak cryptography leave the updated software or firmware vulnerable to threat agents attempting to modify the software or firmware to their advantage.

3.1.2.1 T.UPDATE_COMPROMISE

Threat agents may attempt to provide a compromised update of the software or firmware which undermines the security functionality of the device. Non-validated updates or updates validated using non-secure or weak cryptography leave the update firmware vulnerable to surreptitious alteration.

3.1.3 Audited Activity

Auditing of network device activities is a valuable tool for Administrators to monitor the status of the device. It provides the means for Administrator accountability, security functionality activity reporting, reconstruction of events, and problem analysis. Processing performed in response to device activities may give indications of a failure or compromise of the security functionality. When indications of activity that impact the security functionality are not generated and monitored, it is possible for such activities to occur without Administrator awareness. Further, if records are not generated and retained, reconstruction of the network and the ability to understand the extent of any compromise could be negatively affected.

Additional concerns are the protection of the audit data that is recorded from alteration or unauthorized deletion. This could occur within the TOE, or while the audit data is in transit to an external storage device.

Note [the NDcPPv2.0e] requires that the network device generate the audit data and have the capability to send the audit data to a trusted network entity (e.g., a syslog server).

3.1.3.1 T.UNDETECTED_ACTIVITY

Threat agents may attempt to access, change, and/or modify the security functionality of the network device without Administrator awareness. This could result in the attacker finding an avenue (e.g.,

misconfiguration, flaw in the product) to compromise the device and the Administrator would have no knowledge that the device has been compromised.

3.1.4 Administrator and Device Credentials Data

A network device contains data and credentials which must be securely stored and must appropriately restrict access to authorized entities. Examples include the device firmware, software, configuration authentication credentials for secure channels, and Administrator credentials. Device and Administrator keys, key material, and authentication credentials need to be protected from unauthorized disclosure and modification. Furthermore, the security functionality of the device needs to require default authentication credentials, such as Administrator passwords, be changed.

Lack of secure storage and improper handling of credentials and data, such as unencrypted credentials inside configuration files or access to secure channel session keys, can allow an attacker to not only gain access to the network device, but also compromise the security of the network through seemingly authorized modifications to configuration or though man-in-the middle attacks. These attacks allow an unauthorized entity to gain access and perform administrative functions using the Security Administrator's credentials and to intercept all traffic as an authorized endpoint. This results in difficulty in detection of security compromise and in reconstruction of the network, potentially allowing continued unauthorized access to Administrator and device data.

3.1.4.1 T.SECURITY_FUNCTIONALITY_COMPROMISE

Threat agents may compromise credentials and device data enabling continued access to the network device and its critical data. The compromise of credentials includes replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the Administrator or device credentials for use by the attacker.

3.1.4.2 T.PASSWORD_CRACKING

Threat agents may be able to take advantage of weak administrative passwords to gain privileged access to the device. Having privileged access to the device provides the attacker unfettered access to the network traffic, and may allow them to take advantage of any trust relationships with other network devices.

3.1.5 Device Failure

Security mechanisms of the network device generally build up from roots of trust to more complex sets of mechanisms. Failures could result in a compromise to the security functionality of the device. A network device self-testing its security critical components at both start-up and during run-time ensures the reliability of the device's security functionality.

3.1.5.1 T.SECURITY_FUNCTIONALITY_FAILURE

An external, unauthorized entity could make use of failed or compromised security functionality and might therefore subsequently use or abuse security functions without prior authentication to access, change or modify device data, critical network traffic or security functionality of the device.

3.2 Assumptions

This section describes the assumptions made in identification of the threats and security requirements for network devices. The network device is not expected to provide assurance in any of these areas, and as a result, requirements are not included to mitigate the threats associated.

3.2.1 A.PHYSICAL_PROTECTION

The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the [NDcPPv2.0e] will not include any requirements on physical tamper protection or other physical attack mitigations. The [NDcPPv2.0e] will not expect the product to defend against physical access to the device that allows unauthorized entities to extract data, bypass other controls, or otherwise manipulate the device.

3.2.2 A.LIMITED_FUNCTIONALITY

The device is assumed to provide networking functionality as its core function and not provide functionality/services that could be deemed as general purpose computing. For example, the device should not provide a computing platform for general purpose applications (unrelated to networking functionality).

3.2.3 A.NO_THRU_TRAFFIC_PROTECTION

A standard/generic network device does not provide any assurance regarding the protection of traffic that traverses it. The intent is for the network device to protect data that originates on or is destined to the device itself, to include administrative data and audit data. Traffic that is traversing the network device, destined for another network entity, is not covered by the NDcPPv2.0e. It is assumed that this protection will be covered by cPPs for particular types of network devices (e.g., firewall).

3.2.4 A.TRUSTED_ADMINISTRATOR

The Security Administrator(s) for the network device are assumed to be trusted and to act in the best interest of security for the organization. This includes being appropriately trained, following policy, and adhering to guidance documentation. Administrators are trusted to ensure passwords/credentials have sufficient strength and entropy and to lack malicious intent when administering the device. The network device is not expected to be capable of defending against a malicious Administrator that actively works to bypass or compromise the security of the device.

3.2.5 A.REGULAR_UPDATES

The network device firmware and software is assumed to be updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

3.2.6 A.ADMIN_CREDENTIALS_SECURE

The Administrator's credentials (private key) used to access the network device are protected by the platform on which they reside.

3.2.7 A.RESIDUAL_INFORMATION

The Administrator must ensure that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

3.3 Organizational Security Policy

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. The description of each policy is described in the section below.

3.3.1 P.ACCESS_BANNER

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

4 Security Objectives

The security objectives have been taken from [NDcPPv2.0e] and are reproduced here for the convenience of the reader.

4.1 Security Objectives for the Operational Environment

The following subsections describe objectives for the Operational Environment.

4.1.1 OE.PHYSICAL

Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.

4.1.2 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.

4.1.3 OE.NO_THRU_TRAFFIC_PROTECTION

The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.

4.1.4 OE.TRUSTED_ADMIN

Security Administrators are trusted to follow and apply all guidance documentation in a trusted manner.

4.1.5 OE.UPDATES

The TOE firmware and software is updated by an Administrator on a regular basis in response to the release of product updates due to known vulnerabilities.

4.1.6 OE.ADMIN_CREDENTIALS_SECURE

The Administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

4.1.7 OE.RESIDUAL_INFORMATION

The Security Administrator ensures that there is no unauthorized access possible for sensitive residual information (e.g. cryptographic keys, keying material, PINs, passwords etc.) on networking equipment when the equipment is discarded or removed from its operational environment.

5 Security Requirements

This section identifies the Security Functional Requirements for the TOE and/or Platform. The Security Functional Requirements included in this section are derived from the CC Part 2 and all applicable Protection Profiles as described in section 2.

5.1 Conventions

The CC defines operations on Security Functional Requirements: assignments, selections, assignments within selections and refinements. This document uses the following font conventions to identify the operations defined by the CC:

- Assignment: Indicated with *italicized* text;
- Refinement made by PP author: Indicated with **bold** text and ~~strikethroughs~~, if necessary;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with *italicized and underlined text*;
- Iteration: indicated by adding a string starting with "/" (e.g. "FCS_COP.1/Hash");
- Where operations were completed in the PP itself, the formatting used in the PP has been retained.

Explicitly stated SFRs are identified by having a label 'EXT' after the requirement name for TOE SFRs. Formatting conventions outside of operations matches the formatting specified within the PP.

5.2 TOE Security Functional Requirements

This section identifies the Security Functional Requirements for the TOE. The TOE Security Functional Requirements that appear below in Table 6 are described in more detail in the following subsections.

5.2.1 Class: Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

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

a) Start-up and 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 7.*

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 7*.

Requirement	Auditable Events	Additional Audit Record Contents
Mandatory SFRs		
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.
FCS_COP.1/SigGen	None.	None.
FCS_COP.1/Hash	None.	None.
FCS_COP.1/KeyedHash	None.	None.
FCS_RBG_EXT.1	None.	None.
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
FMT_MOF.1/ManualUpdate	Any attempt to initiate a manual update	None.
FMT_MTD.1/CoreData	All management activities of TSF data.	None.
FMT_SMF.1	None.	None.
FMT_SMR.2	None.	None.
FPT_SKP_EXT.1	None.	None.
FPT_APW_EXT.1	None.	None.
FPT_TST_EXT.1	None.	None.
FPT_TUD_EXT.1	Initiation of update; result of the update attempt (success or failure)	None.
FPT_STM_EXT.1	Discontinuous changes to time – either Administrator actuated or changed via an automated process. (Note that no continuous changes to time need to be logged. See also application note on FPT_STM_EXT.1 in [NDcPPv2.0e])	For discontinuous changes to time: The old and new values for the time. Origin of the attempt to change time for success and failure (e.g., IP address).
FTA_SSL_EXT.1 (if “terminate the session” is selected)	The termination of a local session by the session locking mechanism.	None.
FTA_SSL.3	The termination of a remote session by the session locking mechanism.	None.
FTA_SSL.4	The termination of an interactive session.	None.
FTA_TAB.1	None.	None.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1/Admin	Initiation of the trusted path. Termination of the trusted path. Failure of the trusted path functions.	None.
Optional SFRs		

Requirement	Auditable Events	Additional Audit Record Contents
FMT_MTD.1/CryptoKeys	Management of cryptographic keys.	None.
Selection-Based SFRs		
FCS_HTTPS_EXT.1	Failure to establish a HTTPS session	Reason for failure
FCS_TLSC_EXT.2	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FIA_X509_EXT.1/Rev	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MOF.1/Functions	Modification of the behaviour of the transmission of audit data to an external IT entity, the handling of audit data, the audit functionality when Local Audit Storage Space is full.	None.

Table 6 TOE Security Functional Requirements and Auditable Events

FAU_GEN.2 User identity Association

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

FAU_STG_EXT.1 Protected Audit Event Storage

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

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

FAU_STG_EXT.1.3 The TSF shall [overwrite previous audit records according to the following rule: *[based upon configured threshold]*] when the local storage space for audit data is full.

5.2.2 Class: Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

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

- *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] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;*
- *FFC Schemes using Diffie-Hellman group 14 that meet the following: RFC 3526, Section 3*

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

FCS_CKM.2 Cryptographic Key Establishment

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

- *Elliptic curve-based key establishment schemes that meet the following: NIST Special*

Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";

- *Key establishment scheme using Diffie-Hellman group 14 that meets the following: RFC 3526, Section 3.*

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

FCS_CKM.4 Cryptographic Key Destruction

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

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

that meets the following: *No Standard*.

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

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

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

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

- RSA Digital Signature Algorithm and cryptographic key sizes (**modulus**) [*2048 bits, 3072 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,

].

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*] ~~and cryptographic key sizes [assignment: *cryptographic key sizes*]~~ and message digest sizes [***160, 256, 384***] bits that meet the following: [*assignment: ISO/IEC 10118-3:2004]*.

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

FCS_COP.1.1/KeyedHash The TSF shall perform **keyed-hash message authentication** in accordance with a specified cryptographic algorithm [*HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384*] and

cryptographic key sizes [*160, 256, 284*] **and message digest sizes [*160, 256, 384*] bits** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

FCS_HTTPS_EXT.1 HTTPS Protocol

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

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

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

FCS_RBG_EXT.1 Random Bit Generation

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

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

FCS_TLSC_EXT.2 TLS Client Protocol with authentication

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

[

- *TLS_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_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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_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]*.

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier per RFC 6125 section 6.

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

FCS_TLSC_EXT.2.4 The TSF shall [*not present the Supported Elliptic Curves Extension*] in the Client Hello.

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

FCS_TLSS_EXT.1 TLS Server Protocol

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

- *TLS_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_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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289*
- *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289].*

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

FCS_TLSS_EXT.1.3 The TSF shall [generate EC Diffie-Hellman parameters over NIST curves *[secp256r1, secp384r1] and no other curves; generate Diffie-Hellman parameters of size [2048 bits*]].

5.2.3 Class: Identification and Authentication (FIA)

FIA_AFL.1 Authentication Failure Management

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

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

FIA_PMG_EXT.1 Password Management

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

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

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.

FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2.1 The TSF shall provide a local password-based authentication mechanism, and [*no other authentication mechanism*] to perform local administrative user authentication.

FIA_UAU.7 Protected Authentication Feedback

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

FIA_X509_EXT.1/Rev X.509 Certificate Validation

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

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

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

FIA_X509_EXT.2 X.509 Certificate Authentication

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

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

FIA_X509_EXT.3 X.509 Certificate Requests

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

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

5.2.4 Class: Security Management (FMT)

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, handling of audit data, audit functionality when Local Audit Storage Space is full*] to *Security Administrators*.

FMT_MOF.1/ManualUpdate Management of security functions behavior

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

FMT_MTD.1/CoreData Management of TSF Data

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

FMT_MTD.1/CryptoKeys Management of TSF data

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

FMT_SMF.1 Specification of Management Functions

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

[

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

- *Ability to configure the cryptographic functionality;*
- *Ability to re-enable an Administrator account;*
- *Ability to set the time which is used for time-stamps;*
- *Ability to configure the reference identifier for the peer.]*.

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 the user 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 Class: Protection of the TSF (FPT)

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

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

FPT_APW_EXT.1 Protection of Administrator Passwords

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

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

FPT_TST_EXT.1 TSF Testing

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [*during initial start-up (on power on)*] to demonstrate the correct operation of the TSF: [*Configuration file integrity test, Firmware integrity test, known answer tests, SP 800-90A health tests, RNG known answer test*].

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.

FPT_STM_EXT.1 Reliable Time Stamps

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

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

5.2.6 Class: TOE Access (FTA)

FTA_SSL_EXT.1 TSF-initiated Session Locking

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

- *terminate the session*]

after a Security Administrator-specified time period of inactivity.

FTA_SSL.3 TSF-initiated Termination

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

FTA_SSL.4 User-initiated Termination

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

FTA_TAB.1 Default TOE Access Banners

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

5.2.7 Class: Trusted Path/Channels (FTP)

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 [**audit server**].

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 SFR Dependencies Rationale for SFRs

The Collaborative Protection Profile for Network Devices contains all the requirements claimed in this Security Target. As such, SFR dependencies are not applicable since the PP has been approved.

5.4 Security Assurance Requirements

The TOE assurance requirements for this ST are taken directly from the Collaborative Protection Profile for Network Devices. The assurance requirements are summarized in the table below.

Assurance Class	Components	Components Description
Security Target(ASE)	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Components Definition
	ASE_INT.1	ST Introduction
	ASE_OBJ.1	Security Objectives for the operational environment
	ASE_REQ.1	Stated security requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE Summary Specification
Development(ADV)	ADV_FSP.1	Basic Functional Specification
Guidance documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative User Guidance
Life cycle support(ALC)	ALC_CMC.1	Labelling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests (ATE)	ATE_IND.1	Independent Testing – conformance
Vulnerability Assessment (AVA)	AVA_VAN.1	Vulnerability survey

Table 7 Security Assurance Requirements

5.5 Rationale for Security Assurance Requirements

The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the

interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.

5.6 Assurance Measures

The TOE satisfies the identified assurance requirements. This section identifies the Assurance Measures applied by Fortinet to satisfy the assurance requirements. The table below lists the details.

SAR Component	How the SAR will be met
ADV_FSP.1	The functional specification describes the external interfaces of the TOE; such as the means for a user to invoke a service and the corresponding response of those services. The description includes the interface(s) that enforces a security functional requirement, the interface(s) that supports the enforcement of a security functional requirement, and the interface(s) that does not enforce any security functional requirements. The interfaces are described in terms of their purpose (general goal of the interface), method of use (how the interface is to be used), parameters (explicit inputs to and outputs from an interface that control the behavior of that interface), parameter descriptions (tells what the parameter is in some meaningful way), and error messages (identifies the condition that generated it, what the message is, and the meaning of any error codes). The development evidence also contains a tracing of the interfaces to the SFRs described in this ST.
AGD_OPE.1	The Administrative Guide provides the descriptions of the processes and procedures of how the administrative users of the TOE can securely administer the TOE using the interfaces that provide the features and functions detailed in the guidance.
AGD_PRE.1	The Installation Guide describes the installation, generation, and startup procedures so that the users of the TOE can put the components of the TOE in the evaluated configuration.
ALC_CMC.1	The Configuration Management (CM) documents describe how the consumer identifies the evaluated TOE. The CM documents identify the configuration items, how those configuration items are uniquely identified, and the adequacy of the procedures that are used to control and track changes that are made to the TOE. This includes details on what changes are tracked, how potential changes are incorporated, and the degree to which automation is used to reduce the scope for error.
ALC_CMS.1	
ATE_IND.1	Fortinet will provide the TOE for testing.
AVA_VAN.1	Fortinet will provide the TOE for testing.

Table 8 TOE Security Assurance Measures

6 TOE Summary Specification

This chapter identifies and describes how the Security Functional Requirements identified above are met by the TOE.

TOE SFR	Rationale
FAU_GEN.1	For all administrative actions, including management of the TOE, authentication is required before any actions can occur on the TOE. All administrative actions result in an audit log being generated. These event include, but are not limited to, administrative login and log out, any changes to data as the result of configuration changes, any cryptographic key related activities (for key related events the associated certificate label is recorded in the audit records), and resetting passwords. When an action identified in Table 6 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 logging messages to the audit log for interactions which occur via HTTPS and local console. These events include attempts to establish or terminate sessions and errors detected during decryption or validation of data.
FAU_GEN.2	The TOE ensures that each auditable event is associated with the user that triggered the event. For example, a human user, user identity or related session ID would be included in the audit record. For an IT entity or device, the IP address, MAC address, host name, or other configured identification is included in the audit record.
FAU_STG_EXT.1	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. Only authorized administrators may view these records, and no capability to modify the records is provided. 80% of the appliance disk capacity is reserved for local audit log storage. 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 separate from the locally stored logs accommodating a default of 32K audit records (cache size configurable) 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.
FCS_CKM.1	In support of secure cryptographic protocols, the TOE supports RSA key generation schemes as specified in FIPS PUB 186-4, with key sizes of 2048 and 3072 bits. These keys are used in support of digital certificates for TLS.

	Additionally, in support of the ECDH key exchange, the TOE supports FIPS PUB 186-4 ECC key generation with P-256 and P-384 curves. Finally, Diffie-Hellman group 14 key generation in support of TLS connections is also included.
FCS_CKM.2	The TSF supports two key establishment schemes, as follows: • Diffie-Hellman group 14 • SP 800-56A ECDH key exchange Both key establishment schemes are used in support of TLS communications. The TOE acts as a sender and receiver for all TLS. In support of Diffie-Hellman group 14 the TOE uses prime defined in section 3 of RFC3526.
FCS_CKM.4	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. 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 (zeros) 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.
FCS_COP.1/DataEncryption	The TOE provides symmetric encryption and decryption capabilities using 128 and 256 bit AES in CBC and GCM modes as described in NIST SP 800-38A. AES is implemented in the following protocols: TLS.
FCS_COP.1/SigGen	The TOE provides cryptographic signature generation and verification services using RSA Signature Algorithm with key size of 2048 and greater. These RSA signature verification services are used in the TLS protocols.
FCS_COP.1/Hash	The TOE provides cryptographic hashing services using SHA-1, SHA-256, and SHA-384 as specified in FIPS Pub 180-4 “Secure Hash Standard.” SHS is implemented in the following parts of the TSF: • TLS; • Digital signature verification as part of trusted update validation; and • Hashing of passwords in non-volatile storage.
FCS_COP.1/KeyedHash	The TOE provides keyed-hashing message authentication services using HMAC-SHA-1, HMAC-SHA-256, and HMAC-SHA-384 as specified in FIPS Pub 198-1, "The Keyed-Hash Message Authentication Code,” and FIPS 180-4, “Secure Hash Standard.” HMAC is implemented in the following protocols: TLS.

<table>
<tr><td></td><td>The characteristics of the HMACs used in the TOE are given in the following table:

<table><tr><th>Algorithm</th><th>Hash function</th><th>Block size</th><th>Key size</th><th>Digest size</th></tr><tr><td>HMAC-SHA-1</td><td>SHA-1</td><td>512 bits</td><td>512 bits</td><td>160 bits</td></tr><tr><td>HMAC-SHA-256</td><td>SHA-256</td><td>512 bits</td><td>512 bits</td><td>256 bits</td></tr><tr><td>HMAC-SHA-384</td><td>SHA-384</td><td>1024 bits</td><td>512 bits</td><td>384 bits</td></tr></table></td></tr>
<tr><td>FCS_HTTPS_EXT.1</td><td>The TOE provides management functionality over an HTTPS connection using the TLS implementation described below. 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.</td></tr>
<tr><td>FCS_RBG_EXT.1</td><td>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 6.0 contains a CTR_DRBG implemented per NIST SP 800-90A 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.</td></tr>
<tr><td>FCS_TLSC_EXT.2
FCS_TLSS_EXT.1</td><td>The Network Web-Based GUI uses the HTTPS protocol for secure administrator communications. With respect to the TOE implementation of HTTPS, TLS version 1.1 or 1.2 is used to encrypt and authenticate administration sessions between the remote browser and TOE. All other versions of SSL/TLS are rejected. The TOE supports 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_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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
• TLS_ECDHE_RSA_WITH_AES_256_CBC_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 1.1 or 1.2 is also used for the purposes of protecting the audit logs while in transit to the audit servers. All other versions of SSL/TLS are rejected.

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 or 3072 bit RSA public certificate
• Server generates, signs (RSA PKCS#1) and sends DH parameters and DH public value (2048-bits)
• 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-SHA2-256) the data exchange.</td></tr>
</table>

	The TOE supports ECDH parameters over NIST curves secp256r1 or secp384r1. This support is by default with no configuration required. 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. This is over a mutually authenticated TLS connection. This these connections, the TOE presents a client side certificate which is validated by the external server. The TOE supports reference IDs of hostname (configured by the administrative user). The TOE does not support wildcards or IP addresses. The TOE does not support certificate pinning.
FIA_AFL.1	The TOE provides administrators to specify a maximum number of authentication attempts (between 3 and 5) that can be attempted before a user account is locked out from the remote GUI. 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 does not lock an administrative user out. This prevents a situation where no administrator access is available.
FIA_PMG_EXT.1	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: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")"). The minimum password length is settable by the Authorized Administrator and can range from 8 to 15 characters.
FIA_UIA_EXT.1 FIA_UAU_EXT.2	The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed. Administrative access to the TOE is facilitated through one of several interfaces, • Directly connecting to each TOE appliance • Remotely connecting to appliance GUI via HTTPS/TLS Regardless of the interface at which the administrator interacts, the TOE prompts the user/client for a credential. Only after the administrative user presents the correct authentication credentials will they be granted access to the TOE administrative functionality. No TOE administrative function access is permitted until an administrator is successfully identified and authenticated. The TOE provides a local password based authentication mechanism.

	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 (eg. password). The TOE then either grants administrative access (if the combination of username and credential is correct) or indicates that the login was unsuccessful. The TOE does not provide a reason for failure in the cases of a login failure. The TOE does not permit any administrative function to be accessible until after an administrator is successfully identified and authenticated.
FIA_UAU.7	The TOE obscures all characters entered when attempting password authentication.
FIA_X509_EXT.1/Rev FIA_X509_EXT.2 FIA_X509_EXT.3	The TOE performs X.509 certificate validation at the following points: • TOE TLS client authentication of server X.509 certificates; • When certificates are loaded into the TOE, such as when importing CAs, certificate responses and other device-level certificates (such as the web server certificate presented by the TOE TLS web GUI). In all scenarios, certificates are checked for several validation characteristics: • If the certificate 'notAfter' date is in the past, then this is an expired certificate which is considered invalid; • The certificate chain must terminate with a trusted CA certificate; • Server certificates consumed by the TOE TLS client must have a 'serverAuthentication' extendedKeyUsage purpose; • Client certificates consumed by the TOE TLS server (for mutual authentication) must have a 'clientAuthentication' extendedKeyUsage purpose; A trusted CA certificate is defined as any certificate loaded into the TOE trust store that has, at a minimum, a basicConstraints extension with the CA flag set to TRUE. Certificate revocation checking is performed using CRL. The TOE has a trust store where root CA and intermediate CA certificates can be stored. The trust store is not cached: if a certificate is deleted, it is immediately untrusted. If a certificate is added to the trust store, it is immediately trusted for its given scope. If, during the entire trust chain verification activity, any certificate under review fails a verification check, then the entire trust chain is deemed untrusted and the TLS connection is terminated. As part of the verification process, CRL is used to determine whether the certificate is revoked or not. If the CRL service cannot be contacted, then the TOE will choose to automatically reject the certificate in this case. Instructions for configuring the trusted IT entities to supply appropriate X.509 certificates are captured in the guidance documents.
FMT_MOF.1/Functions	The TOE restricts the ability to configure Syslog to the Admin role.
FMT_MOF.1/ManualUpdate	The TOE restricts the ability to perform software updates to the Admin role.

FMT_MTD.1/CoreData	The TOE provides Security Administrators with the ability to access the TOE via local CLI and TLS in order to configure and view audit data, configuration data, security values, user accounts, and trusted updates.
FMT_MTD.1/CryptoKeys	The TOE restricts the ability to manage TLS and any configured X.509 private keys to the Admin role.
FMT_SMF.1 FMT_SMR.2	The TOE may be managed via the CLI (console) or GUI (HTTPS). The specific management capabilities include: • Ability to administer the TOE locally and remotely (GUI & CLI); • Ability to configure the access banner (GUI & CLI); • Ability to configure the session inactivity time before session termination or locking (GUI & CLI); • Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates (CLI & GUI); • Ability to configure the authentication failure parameters (CLI); • Ability to configure audit behavior (CLI); • Ability to configure the cryptographic functionality (GUI & CLI); • Ability to set the time which is used in time-stamps (GUI & CLI) • Ability to configure login banner • Ability to re-enable admin account
FPT_SKP_EXT.1	The TOE stores all private keys in a secure directory that is not readily accessible to administrators; hence no interface access.
FPT_APW_EXT.1	The TOE stores Security Administrator passwords. All passwords are stored in a secure directory that is not readily accessible to administrators. The passwords are hashed and not in plaintext.
FPT_TST_EXT.1	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 integrity test: The configuration file integrity test is run automatically at startup. A 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 digital signature verification that the image file that's about to be loaded was properly signed and has maintained its integrity since being signed. • AES, SHA, 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. Indication of successful tests would appear as follows: Running <test>... passed

	Completion of all self-tests is indicated by: Self-tests passed 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. When these tests are run, it is confirmed that both the image and configuration file are operating in a known good state. These tests also verify that the cryptographic algorithms operate correctly and will not inadvertently release plaintext data.
FPT_TUD_EXT.1	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: • 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.
FPT_STM_EXT.1	The TOE provides a source of date and time information used in audit event timestamps. The clock function is reliant on the system clock provided by the underlying hardware. This date and time is used as the time stamp that is applied to TOE-generated audit records and used to track inactivity of administrative sessions.
FTA_SSL_EXT.1 FTA_SSL.3	A Security Administrator can configure maximum inactivity times for administrative sessions through the TOE GUI and CLI interfaces. The configuration of inactivity periods is applied on a per interface basis. 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.
FTA_SSL.4	A Security Administrator is able to exit out of both local and remote administrative sessions.
FTA_TAB.1	Security Administrators can define a custom login banner that will be displayed at the following interfaces, • Local CLI • Remote GUI This banner will be displayed prior to allowing Security Administrator access through those interfaces.

FTP_ITC.1	The TOE supports communications with several types of authorized IT entities, including, • Audit Server Each of these connections are protected via a TLS connection. This protects the data from disclosure by encryption using AES and by HMACs that verify that data has not been modified. 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.
FTP_TRP.1/Admin	All remote administrative communications take place over a secure encrypted session. Remote GUI connections take place over a TLS connection. The TLS session is encrypted using AES encryption and uses HMACs to protect integrity. The remote administrators can initiate TLS communications with the TOE.

Table 9 TOE Summary Specification SFR Description

6.1 Key Storage and Zeroization

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.

Key or CSP	Storage	Zeroization Method
Firmware Update Key	Flash storage (PT)	Format flash storage (overwritten with zeros)
Firmware Integrity Key	Flash storage (PT)	Format flash storage (overwritten with zeros)
HTTPS/SSL Server/Host Key	Flash storage (PT)	Format flash storage (overwritten with zeros)
HTTPS/TLS Session Authentication Key	RAM	Power cycle or reboot; session terminated (overwritten with zeros)
HTTPS/TLS Session Encryption Key	RAM	Power cycle or reboot; session terminated (overwritten with zeros)
Configuration Integrity Key	Flash storage (PT)	Format flash storage (overwritten with zeros)
Configuration Encryption Key	RAM	Power cycle or reboot (overwritten with zeros)
Configuration Backup Key	Flash storage (PT)	Format flash storage (overwritten with zeros)
Operator Password	Flash storage (hashed)	Factory reset (overwritten with zeros)
User Password	Flash storage (hashed)	Factory reset (overwritten with zeros)

Table 10 **Key Storage & Zeroization**

Annex A: References

The following documentation was used to prepare this ST:

Identifier	Description
[CC_PART1]	Common Criteria for Information Technology Security Evaluation – Part 1: Introduction and general model, dated September 2012, version 3.1, Revision 4
[CC_PART2]	Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated September 2012, version 3.1, Revision 4
[CC_PART3]	Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components September 2012, version 3.1, Revision 4
[CEM]	Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated September 2012, version 3.1, Revision 4
[NDcPPv2.0e]	collaborative Protection Profile for Network Devices (NDcPP) + Errata 20180314 version 2.0e March 14, 2018
[SD]	Supporting Document Mandatory Technical Document: Evaluation Activities for Network Device cPP, Version 2.0e, March 14, 2018.
[800-38A]	NIST Special Publication 800-38A Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001
[800-38D]	NIST Special Publication 800-38D Recommendation for Block Cipher Modes of Operation: Galois/Counter Mode (GCM) and GMAC, November 2007.
[800-56Ar2]	NIST Special Publication 800-56A Revision 2, May 2013, Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography
[800-56B]	NIST Special Publication 800-56B Revision 1, September 2014, Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography
[FIPS 140-2]	FIPS PUB 140-2 Federal Information Processing Standards Publication Security Requirements for Cryptographic Modules May 25, 2001
[FIPS PUB 186-4]	FIPS PUB 186-4 Federal Information Processing Standards Publication: Digital Signature Standard (DSS), July 2013.
[FIPS PUB 198-1]	FIPS PUB 198-1 Federal Information Processing Standards Publication: The Keyed-Hash Message Authentication Code (HMAC) July 2008
[800-90A]	NIST Special Publication 800-90A Revision 1, Recommendation for Random Number Generation Using Deterministic Random Bit Generators, June 2015
[FIPS PUB 180-4]	FIPS PUB 180-4 Federal Information Processing Standards Publication Secure Hash Standard (SHS), August 2015.
[RFC3526]	RFC 3526, More Modular Exponential (MODP) Diffie-Hellman groups for Internet Key Exchange (IKE), May 2003.
[RFC2818]	RFC 2818, HTTP Over TLS, May 2000.
[RFC5647]	RFC 5647, AES Galois Counter Mode for the Secure Shell Transport Layer Protocol, August 2009.
[RFC5246]	RFC 5246, The Transport Layer Security (TLS) Protocol Version 1.2, August 2008.
[RFC4346]	RFC 4346, The Transport Layer Security (TLS) Protocol Version 1.1, April 2006.
[RFC3268]	RFC 3268, Advanced Encryption Standard (AES) Ciphersuites for Transport Layer Security (TLS), June 2002.
[RFC5289]	RFC 5289, TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM), August 2008.

[RFC6125]	RFC 6125, Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS), March 2011.
[RFC5280]	RFC 5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008.
[RFC2986]	RFC 2986, PKCS #10: Certification Request Syntax Specification Version 1.7, November 2000.

Table 11: References