Cisco Email Security Appliance

Security Target

Version 1.0

September 13, 2017

Table of Contents

List of Tables

List of Figures

Acronyms

The following acronyms and abbreviations are common and may be used in this Security Target:

Table 1 Acronyms

Acronyms / Abbreviations	Definition
ACL	Access Control Lists
AES	Advanced Encryption Standard
BRI	Basic Rate Interface
CC	Common Criteria for Information Technology Security Evaluation
CEM	Common Evaluation Methodology for Information Technology Security
CM	Configuration Management
DHCP	Dynamic Host Configuration Protocol
DSU	Data Service Unit
HTTP	Hyper-Text Transport Protocol
HTTPS	Hyper-Text Transport Protocol Secure
ICMP	Internet Control Message Protocol
ESA	Email Security Appliance
IT	Information Technology
MTA	Mail Transfer Agent
MX	Mail Exchange
NDcPP	Network Device Protection Profile
OS	Operating System
PoE	Power over Ethernet
PP	Protection Profile
SA	Security Association
SHS	Secure Hash Standard
SIO	Cisco Security Intelligence
SIP	Session Initiation Protocol
SSHv2	Secure Shell (version 2)
ST	Security Target
TCP	Transport Control Protocol
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Function
TSP	TOE Security Policy
UDP	User datagram protocol
WAN	Wide Area Network

Terminology

Table 2 Terminology

Term	Definition
Authorized Administrator	Any user which has been assigned to a privilege level that is permitted to perform all TSF-related functions.
Role	An assigned role gives a user varying access to the management of the TOE.
Security Administrator	Synonymous with Authorized Administrator for the purposes of this evaluation.
User	Any entity (human user or external IT entity) outside the TOE that interacts with the TOE.
Vty	vty is a term used by Cisco to describe a single terminal (whereas Terminal is more of a verb or general action term).

DOCUMENT INTRODUCTION

Prepared By:
Cisco Systems, Inc.
170 West Tasman Dr.
San Jose, CA 95134

This document provides the basis for an evaluation of a specific Target of Evaluation (TOE), the Email Security Appliance (ESA). This Security Target (ST) defines a set of assumptions about the aspects of the environment, a list of threats that the product intends to counter, a set of security objectives, a set of security requirements, and the IT security functions provided by the TOE which meet the set of requirements. Administrators of the TOE will be referred to as administrators, Authorized Administrators, TOE administrators, semi-privileged, privileged administrators, and security administrators in this document. The Common Criteria Functional Specification is met through the description of interfaces in this Security Target and the parameters described within the Common Criteria Guidance Documentation as well as the Cisco documentation for ESA.

1 SECURITY TARGET INTRODUCTION

The Security Target contains the following sections:

- Security Target Introduction [Section 1]
- Conformance Claims [Section 2]
- Security Problem Definition [Section 3]
- Security Objectives [Section 4]
- IT Security Requirements [Section 5]
- TOE Summary Specification [Section 6]

The structure and content of this ST comply with the requirements specified in the Common Criteria (CC), Part 1, Annex A, and Part 2.

1.1 ST and TOE Reference

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

Table 3 ST and TOE Identification

Name	Description
ST Title	Email Security Appliance Security Target
ST Version	1.0
Publication Date	September 13, 2017
Vendor and ST Author	Cisco Systems, Inc.
TOE Reference	Email Security Appliance
TOE Hardware Models	C170, C190, C370, C380, C390, C670, C680, C690, C690X, X1070, C100v, C300v, C600v
TOE Software Version	AsyncOS 11.0
Keywords	Email, Data Protection, Authentication, Network Device

1.2 TOE Overview

The TOE, Cisco Email Security Appliance, is a network device. ESA is an appliance that provides comprehensive email protection services for email. It is an email protection product that monitors Simple Mail Transfer Protocol (SMTP) network traffic, analyzes the monitored network traffic using various techniques, and reacts to identified threats associated with email messages (such as spam and inappropriate or malicious content). The TOE includes the hardware models as defined in Table 3 in section 1.1. ESA was evaluated as a network device only and the email protection services were not assessed during this evaluation.

1.2.1 TOE Product Type

Cisco ESA is an email protection product that can block spam, and threats that may be delivered via email. ESA receives updates from the Cisco Security Intelligence (SIO) organization. Cisco SIO prevents zero-hour attacks by continually generating new rules that feed updates to the Cisco ESA. The updates occur every 3 to 5 minutes keeping the ESA threat database updated for current email threats.

Once a threat is detected through email scanning, the ESA will take action based on authorized administrator configurable filters. Email encryption can also be applied to outbound emails.

The Cisco ESA is designed to serve as the SMTP gateway or Mail Exchanger (MX), providing the Message Transfer Agent (MTA) role in the customer's network infrastructure. As such, the ESA should be installed between an external and an internal network, such that network traffic sent and received on TCP port 25[1] must pass through the ESA. ESA provides separate physical interfaces allowing it to be connected to separate internal and external networks. It can be configured to monitor email network traffic sent from the internal network to the external network, and vice versa.

The TOE provides two management interfaces: Command Line Interface (CLI) and web-based Graphical User Interface (GUI). The GUI contains most of the functionality to configure and monitor the system. However, not all CLI commands are available in the GUI; some features are *only* available through the CLI.

ESA provides capabilities to manage its monitoring, analysis, and reaction functions, and controls access to those capabilities through the use of administrative roles with varying security management authorizations. All administrative users of the TOE are required to be identified and authenticated before accessing the TOE's management capabilities. In addition, all security relevant administrative actions are audited.

1.2.2 Supported non-TOE Hardware/ Software/ Firmware

The TOE supports (in some cases optionally) the following hardware, software, and firmware in its environment when the TOE is configured in its evaluated configuration:

Table 4 IT Environment Components

Component	Required	TOE Interface	Usage/Purpose Description for TOE performance
Management Workstation with SSH Client	Yes	Management Port	This includes any IT Environment Management workstation with a SSH client installed that is used by the TOE administrator to support TOE administration through SSH protected channels. Any SSH client that supports SSHv2 may be used.
Management Workstation using web browser for HTTPS	Yes	Management Port	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. Any web browser that supports TLSv1.0 with the supported ciphersuites may be used.
Local Console	No	Serial Console Port	This includes any IT Environment Console that is directly connected to the TOE via the Serial Console Port and is used by the TOE administrator to support TOE administration.
NTP Server	No	Management Port	The TOE supports communications with an NTP server in order to synchronize the date and time on the TOE with the NTP server's date and time. A solution must be used that supports secure communications with up to a 32 character key.
SMTP Server	Yes	10/100/1000 Port	This includes any SMTP servers that the TOE receives email from and sends email to.
Syslog Server	Yes	Management Port	This includes any syslog server to which the TOE would transmit syslog messages.

[1] SMTP traffic typically is communicated on TCP port 25, but the TOE can be configured to monitor other ports for SMTP traffic.

Component	Required	TOE Interface	Usage/Purpose Description for TOE performance
Update Server	No	Management Port	This includes the Cisco IT environment update servers that are used to download the latest software updates for the TOE.

1.3 TOE DESCRIPTION

This section provides an overview of the Cisco Email Security Appliance TOE. ESA is a security appliance that scans traffic between an external network and the customer's internal network. Traffic flowing to and from the external network to the internal network is first routed through the ESA appliance. The TOE was evaluated as a network device only and the email protection services were not assessed during this evaluation.

Through the intercept, scanning, and reporting functions, the ESA appliance can detect potentially malicious files of various types, filter traffic for restricted content, and email containing spam messages or phishing attempts.

ESA supports RFC 2821-compliant Simple Mail Transfer Protocol (SMTP) to accept and deliver messages.

Cisco ESA monitors SMTP network traffic and applies the following traffic analysis mechanisms:

- Signature analysis - the administrator can configure message filters, comprising rules describing how to handle messages and attachments as they are received. Filter rules identify messages based on message or attachment content, information about the network, message headers, or message body.
- Detection of spam - ESA implements a layered mechanism to detecting and handling spam. The first layer of spam control is called reputation filtering, which allows for classifying email senders and restricting access to email infrastructures based on a sender's trustworthiness as determined by the ESA. The second layer comprises scanning of messages by the ESA's Anti-Spam engine. In addition, the administrator can create policies to deliver messages from known or highly reputable senders directly to the end user without any anti-spam scanning, while messages from less reputable or unknown senders are subjected to anti-spam scanning. ESA can also be configured to throttle the number of messages it will accept from suspicious senders, reject connections or bounce messages.
- Anti-virus scanning - ESA incorporates both Sophos and McAfee Anti-Virus virus scanning engines, which can be configured to scan messages and attachments for viruses on a per-mail policy basis and take the following actions based on the scan results: attempt to repair the attachment; drop the attachment; modify the subject header; add an additional header; send the message to a different address or mail host; archive the message; or delete the message.
- Application of content filters - the administrator can create content filters to be applied to messages on a per-recipient or per-sender basis. Content filters are similar to the message filters described above under "Signature analysis", except that they are applied later in the

email processing pipeline. Email messages can be quarantined, deleted, or have the flagged content filtered from the email. The action taken on the email is based on the content filtering policies configured by the authorized administrator.

- Application of virus outbreak filters - ESA has the ability to compare incoming messages with administrator-configured Virus Outbreak Rules. Messages that match such rules are assigned a threat level and that threat level is compared to the threat level threshold set by the administrator. Messages meeting or exceeding the threshold are quarantined.

Once a suspected infected email or phishing attempt is detected, ESA can then take one or more of the following actions in response as identified by the traffic analysis mechanisms:

- Generate an email to an administrator containing an alarm
- Generate an alarm that is written to a log file that can be examined using the administrator console
- Drop the email message
- Bounce the email message
- Archive the email message
- Add a blind-carbon copied recipient to the email message
- Modify the email message.

The various administrator-configurable rule sets that control the behavior of spam detection, anti-virus scanning, content filtering and virus outbreak filtering are configured such that they are applied to specific groups of users based on email message attributes (Envelope Recipients, Envelope Sender, From: header, or Reply-To: header) in order to perform each type of analysis as described above.

The following figure provides a visual depiction of an example TOE deployment. The TOE boundary is surrounded with a hashed red line.

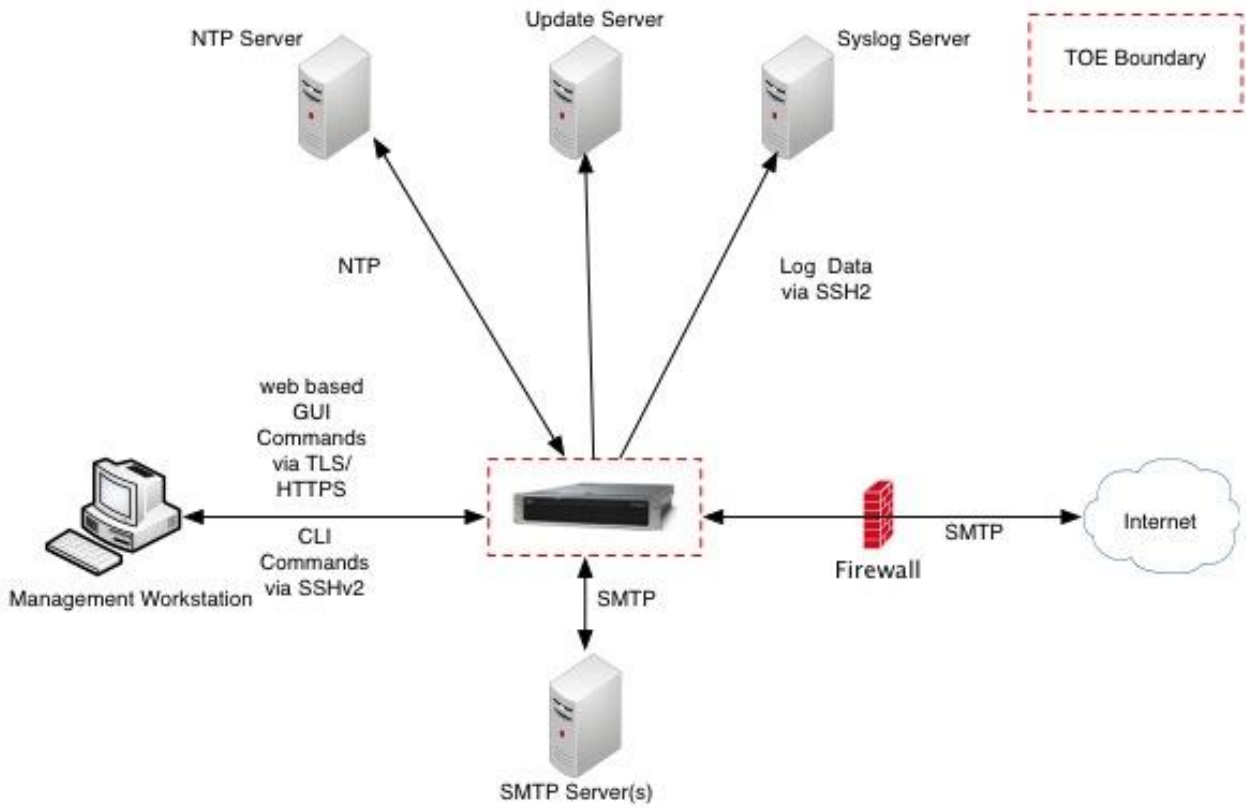

Figure 1 TOE Example Deployment

The previous figure includes the following:

- TOE
- The following are considered to be in the IT Environment:
  - Management Workstation
  - NTP Server
  - Syslog Server
  - SMTP Server
  - Update Server

1.4 TOE Evaluated Configuration

The TOE consists of one or more appliances as specified in section 1.5 below and includes the Cisco AsyncOS software version 11.0. The Cisco AsyncOS configuration determines how packets are handled to and from the TOE's network interfaces. In addition, the appliance configuration determines how suspected malicious email is handled.

The TOE can optionally connect to an NTP server on its internal network for time services. Also, if the ESA is to be remotely administered, then the management station must be connected to an internal network, SSHv2 must be used to remotely connect to the appliance. A syslog server is also used to store audit records. If these servers are used, they must be attached to the internal (trusted) network. The internal (trusted) network is meant to be separated effectively from unauthorized individuals and user traffic; one that is in a controlled environment where implementation of security policies can be enforced.

1.5 Physical Scope of the TOE

The TOE is comprised of both software and hardware. The hardware is comprised of the following: C170, C190, C370, C380, C390, C670, C680, C690, C690X, X1070, C100v, C300v, C600v running on Cisco UCS servers (blade or rack-mounted). The software version of the TOE is Cisco AsyncOS version 11.0.

The Cisco Email Security Appliance that comprises the TOE has common hardware characteristics. These characteristics affect only non-TSF relevant functions of the appliances (such as throughput and amount of storage) and therefore support security equivalency of the appliances in terms of hardware.

The C100v, C300v, and C600v models running on Cisco UCS servers have similar disk layouts, queue and cache sizes, and configurations as their dedicated hardware appliance counterparts. The software images for the C100v, C300v, and C600v have been pre-configured with disk space, queue/cache space, memory, and processor cores. These differences in the pre-configurations of these models are the reason the software images differ.

The TOE guidance documentation that is considered to be part of the TOE can be found listed in the Cisco Email Security Appliance Common Criteria Operational User Guidance and Preparative Procedures document and are downloadable from the http://cisco.com web site. The TOE is comprised of the following specifications as described in Table 5 below:

Table 5 TOE Models and Specifications

Model	Processor	Memory	Hard Disk	Interfaces/UCS Servers
C690	Intel Xeon	32 GB	2.4 TB (600 x 4), RAID 10	(2) USB Console Port (1) Console Port (RJ-45 connector) (1) Management Port (4) 10/100/1000 Port (2) Power Supply (1) Remote Power Management Port
C690X	Intel Xeon 2x2.4 GHz (6 core)	32 GB	2.4 TB (600 x 4), RAID 10	(2) USB Console Port (1) Console Port (RJ-45 connector) (1) Management Port (4) 10/100/1000 Port (2) Power Supply (1) Remote Power Management Port
C680	Intel Xeon 2x2.0 GHz (6 core)	32 GB	1.8 TB (300 x 6), RAID 10	(2) USB Port (1) Serial Console Port (RJ-45 connector) (1) Management Port (3) 10/100/1000 Port (2) Power Supply (1) Remote Power Management Port
C670	Intel Xeon 1x2.4 GHz (2 core)	32 GB	70GB	(2) USB Port

Model	Processor	Memory	Hard Disk	Interfaces/UCS Servers
				(1) Serial Console Port (RJ-45 connector) (1) Management Port (3) 10/100/1000 Port (2) Power Supply (1) Remote Power Management Port
C390	Intel Xeon 1x2.4 GHz (6 core)	16 GB	1.2 TB (600 x 2), RAID 1	(2) USB Console Port (1) Console Port (1) Management Port (3) 10/100/1000 Port (2) Power Supply (1) Remote Power Management Port
C380	Intel Xeon 1x2.0 GHz (6 core)	16 GB	1.2 TB (600 x 2), RAID 1	(2) USB Console Port (1) Console Port (1) Management Port (4) 10/100/1000 Port (2) Power Supply (1) Remote Power Management Port
C370	Intel Xeon 1x2.0 GHz (6 core)	16 GB	1.2 TB (600 x 2), RAID 1	(2) USB Console Port (1) Console Port (1) Management Port (4) 10/100/1000 Port (2) Power Supply (1) Remote Power Management Port
C190	Intel Xeon (1x1.9 GHz, 6 core)	8 GB	250 GB RAID 1	(2) USB Console Port (1) Console Port (1) Management Port (2) 10/100/1000 Port (1) Power Supply(1) Remote Power Management Port
C170	Intel Xeon 1x2.8 GHz (1 Dual core)	4 GB	500 GB RAID 1	(2) USB Console Port (1) Console Port (1) Management Port (2) 10/100/1000 Port (1) Power Supply
X1070	Intel Xeon 2x2.4 GHz (6 core)	32 GB	2.4 TB (600 x 4), RAID 10	(2) USB Console Port (1) Console Port (RJ-45 connector) (1) Management Port (4) 10/100/1000 Port (2) Power Supply (1) Remote Power Management Port
C100v	UCS C-Series[2] running ESXi 5.5	UCS C-Series running ESXi 5.5	UCS C-Series running ESXi 5.5	UCS C-Series running ESXi 5.5
C300v	UCS C-Series running ESXi 5.5	UCS C-Series running ESXi 5.5	UCS C-Series running ESXi 5.5	UCS C-Series running ESXi 5.5
C600v	UCS C-Series running ESXi 5.5	UCS C-Series running ESXi 5.5	UCS C-Series running ESXi 5.5	UCS C-Series running ESXi 5.5

[2] See the UCS C-Series data sheets for details on the interfaces

1.6 Logical Scope of the TOE

The TOE is comprised of several security features. Each of the security features identified above consists of several security functionalities, as identified below.

1. Security Audit
2. Cryptographic Support
3. Identification and Authentication
4. Security Management
5. Protection of the TSF
6. TOE Access
7. Trusted Path/Channels

These features are described in more detail in the subsections below. In addition, the TOE implements all RFCs of the NDcPP v1.0 as necessary to satisfy testing/assurance measures prescribed therein.

1.6.1 Security Audit

The Cisco Email Security Appliance provides extensive auditing capabilities. The TOE can audit events related to cryptographic functionality, identification and authentication, and administrative actions. The Cisco Email Security Appliance generates an audit record for each auditable event. Each security relevant audit event has the date, timestamp, event description, and subject identity. The administrator configures auditable events, performs back-up operations, and manages audit data storage. The TOE provides the administrator with a circular audit trail or a configurable audit trail threshold to track the storage capacity of the audit trail. Audit logs are backed up over an encrypted channel to an external audit server.

1.6.2 Cryptographic Support

The TOE provides cryptography in support of other Cisco ESA security functionality. This cryptography has been validated for conformance to the requirements of FIPS 140-2 Level 1 (see Table 6 for certificate references). The entropy source provides 256 bits of entropy used to seed the RNG. After cryptographic keys are used, they are zeroized. See CiscoSSL FIPS Object Module 6.0 cert #2505 is in use.

Table 6 CAVP References

Algorithm	Cert. #	SFR
AES	4561, 4680	FCS_SSHC_EXT.1 FCS_SSHS_EXT.1 FCS_TLSS_EXT.1 FCS_HTTPS_EXT.1 FCS_COP.1.1(1)
HMAC	3013, 3096	FCS_SSHC_EXT.1 FCS_SSHS_EXT.1 FCS_TLSS_EXT.1 FCS_HTTPS_EXT.1 FCS_COP.1.1(4)
DRBG	1509, 1583	FCS_SSHC_EXT.1 FCS_SSHS_EXT.1 FCS_TLSS_EXT.1 FCS_HTTPS_EXT.1 FCS_RBG_EXT.1
ECDSA	1113, 1155	FCS_TLSS_EXT.1 FCS_HTTPS_EXT.1 FCS_CKM.1 FCS_CKM.2
RSA	2488, 2553	FCS_SSHC_EXT.1 FCS_SSHS_EXT.1 FCS_TLSS_EXT.1 FCS_HTTPS_EXT.1 FCS_CKM.1 FCS_CKM.2 FCS_COP.1.1(2)
SHA	3739, 3831	FCS_SSHC_EXT.1 FCS_SSHS_EXT.1 FCS_TLSS_EXT.1 FCS_HTTPS_EXT.1 FCS_COP.1.1(3)
CVL	1244, 1325	FCS_SSHC_EXT.1 FCS_SSHS_EXT.1 FCS_TLSS_EXT.1 FCS_CKM.2

The DH related certificates apply to the key derivation function used in TLS and SSH.

The TOE provides cryptography in support of remote administrative management via SSHv2. The cryptographic services provided by the TOE are described in Table 7 below.

Table 7 TOE Provided Cryptography

Cryptographic Method	Use within the TOE
Secure Shell Establishment (SSH)	Used to establish initial SSH session
Transport Layer Security (TLS)	Used in TLS session establishment
AES	Used to encrypt TLS session traffic Used to encrypt SSH session traffic
ECDH	Used to provide key exchange in SSH
DH	Used in the key derivation for SSH and TLS

Cryptographic Method	Use within the TOE
RSA Signature Services	Used in TLS session establishment Used in SSH session establishment X.509 certificate signing
HMAC	Used for keyed hash, integrity services in TLS an SSH session establishment
DRBG	Used for random number generation Used in TLS session establishment Used in SSH session establishment
SHA	Used to provide TLS traffic integrity verification

The following SFRs are supported by the Cryptographic Module:

- FCS_CKM.1
- FCS_CKM.2
- FCS_CKM.4
- FCS_COP.1 including all iterations
- FCS_HTTPS_EXT.1
- FCS_SSHS_EXT.1
- FCS_TLSS_EXT.1

1.6.3 Identification and Authentication

The TOE performs two types of authentication: device-level authentication of remote Message Transfer Agents (MTA) and user authentication for the Authorized Administrator of the TOE. Device-level authentication allows the TOE to establish a secure channel with another MTA over TLS. The secure channel is established only after each device authenticates the other with a X.509v3 certificate.

The TOE provides authentication services for administrative users wishing to connect to the TOE's secure CLI and GUI administrative interfaces. Prior to an administrator logging in, a login banner is presented at both the CLI and GUI. The TOE requires Authorized Administrators to authenticate prior to being granted access to any of the management functionality.

The TOE can be configured to require a minimum password length of 15 characters as well as mandatory password complexity rules that includes special characters. The TOE provides administrator authentication against a local user database. Password-based authentication can be performed on the serial console or remote interfaces. The SSHv2 interface also supports authentication using SSH keys. The remote GUI is protected using TLS.

1.6.4 Security Management

The TOE provides secure administrative services for management of general TOE configuration and the security functionality provided by the TOE. All TOE administration occurs either through a secure SSHv2 session or via a local console connection. The TOE provides the ability to securely manage:

- All TOE administrative users;
- All identification and authentication;
- All audit functionality of the TOE;
- All TOE cryptographic functionality;
- The timestamps maintained by the TOE;
- Update to the TOE; and
- TOE configuration file storage and retrieval.

The TOE provides capabilities to manage its security functions, and controls access to those capabilities through the use of administrative roles with varying security management authorizations.

Administrators can create configurable login banners to be displayed at time of login, and can also define an inactivity timeout for each admin interface to terminate sessions after a set period of inactivity.

1.6.5 Protection of the TSF

The TOE protects against interference and tampering by untrusted subjects by implementing identification, authentication, and access controls to limit configuration to Authorized Administrators. The TOE prevents reading of cryptographic keys and passwords.

The TOE internally maintains the date and time. This date and time is used as the timestamp that is applied to audit records generated by the TOE. Administrators can update the TOE's clock manually, or can optionally configure the TOE to use NTP to synchronize the TOE's clock with an external time source. Finally, the TOE performs testing to verify correct operation of the appliance itself and that of the cryptographic module.

The TOE is able to download software updates from the Update Server. The TOE is able to verify any software updates prior to the software updates being installed on the TOE to avoid the installation of unauthorized software.

1.6.6 TOE Access

The TOE can terminate inactive sessions after an Authorized Administrator configurable time-period. Once a session has been terminated the TOE requires the user to re-authenticate to establish a new session.

The TOE can also display an Authorized Administrator specified banner on the CLI and GUI management interfaces prior to allowing any administrative access to the TOE. Administrators are able to exit their own administrator sessions.

1.6.7 Trusted path/Channels

The TOE allows trusted paths to be established to itself from remote administrators over SSHv2 for remote CLI access and HTTPS for remote GUI access. The TOE can push log files to an external syslog server using SCP over SSH.

1.7 Excluded Functionality

The following functionality is excluded from the evaluated configuration.

Table 8 Excluded Functionality

Excluded Functionality	Exclusion Rationale
Non-FIPS 140-2 mode of operation	This mode of operation includes non-FIPS allowed operations.

These services will be disabled by configuration. The exclusion of this functionality does not affect compliance to the collaborative Protection Profile for Network Devices Version 1.0.

Please note that the functionality evaluated is scoped exclusively to the security functional requirements specified in the Section 5 of this document. The email protection services functionality included in the product and described in Section 1.2 and 1.3 were not assessed as part of this evaluation. All other functionality provided by the devices needs to be assessed separately and no further conclusions can be drawn about their effectiveness.

2 CONFORMANCE CLAIMS

2.1 Common Criteria Conformance Claim

The TOE and ST are compliant with the Common Criteria (CC) Version 3.1, Revision 4, dated: September 2012. For a listing of Assurance Requirements claimed see section 5.5.

The TOE and ST are CC Part 2 extended and CC Part 3 conformant.

2.2 Protection Profile Conformance

The TOE and ST are conformant with the Protection Profiles as listed in Table 9 below:

Table 9 Protection Profiles

Protection Profile	Version	Date
collaborative Protection Profile for Network Devices (NDcPP)	1.0	27-Feb-2015

2.3 Protection Profile Conformance Claim Rationale

2.3.1 TOE Appropriateness

The TOE provides all of the functionality at a level of security commensurate with that identified in the U.S. Government Protection Profile:

- collaborative Protection Profile for Network Devices, Version 1.0

2.3.2 TOE Security Problem Definition Consistency

The Assumptions, Threats, and Organizational Security Policies included in the Security Target represent the Assumptions, Threats, and Organizational Security Policies specified in the collaborative Protection Profile for Network Devices Version 1.0 for which conformance is claimed verbatim. All concepts covered in the Protection Profile Security Problem Definition are included in the Security Target Statement of Security Objectives Consistency.

The Security Objectives included in the Security Target represent the Security Objectives specified in the NDcPPv1.0 for which conformance is claimed verbatim. All concepts covered in the Protection Profile's Statement of Security Objectives are included in the Security Target.

2.3.3 Statement of Security Requirements Consistency

The Security Functional Requirements included in the Security Target represent the Security Functional Requirements specified in the NDcPPv1.0 for which conformance is claimed verbatim. All concepts covered in the Protection Profile's Statement of Security Requirements are included in this Security Target. Additionally, the Security Assurance Requirements included in this Security Target are identical to the Security Assurance Requirements included in section 4.3 of the NDcPPv1.0.

3 SECURITY PROBLEM DEFINITION

This chapter identifies the following:

- ♦ Significant assumptions about the TOE's operational environment.
- ♦ IT related threats to the organization countered by the TOE.
- ♦ Environmental threats requiring controls to provide sufficient protection.
- ♦ Organizational security policies for the TOE as appropriate.

This document identifies assumptions as A.assumption with "assumption" specifying a unique name. Threats are identified as T.threat with "threat" specifying a unique name. Organizational Security Policies (OSPs) are identified as P.osp with "osp" specifying a unique name.

3.1 Assumptions

The specific conditions listed in the following subsections are assumed to exist in the TOE's environment. These assumptions include both practical realities in the development of the TOE security requirements and the essential environmental conditions on the use of the TOE.

Table 10 TOE Assumptions

Assumption	Assumption Definition
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 cPP will not include any requirements on physical tamper protection or other physical attack mitigations. The cPP 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.
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 computing platform for general purpose applications (unrelated to networking functionality).
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 ND cPP. It is assumed that this protection will be covered by cPPs for particular types of network devices (e.g, firewall).
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

Assumption	Assumption Definition
	malicious administrator that actively works to bypass or compromise the security of the device.
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.
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 Threats

The following table lists the threats addressed by the TOE and the IT Environment. The assumed level of expertise of the attacker for all the threats identified below is Enhanced-Basic.

Table 11 Threats

Threat	Threat Definition
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.
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.
T.UNTRUSTED_COMMUNICATION_CHANNELS	Threat agents may attempt to target network devices that do not use standardized secure tunneling protocols to protect the critical network traffic. Attackers may take advantage of poorly designed protocols or poor key management to successfully perform man-in-themiddle 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.

Threat	Threat Definition
T.WEAK_AUTHENTICATION_ENDPOINTS	Threat agents may take advantage of secure protocols that use weak methods to authenticate the endpoints – e.g., 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.
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.
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.
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 include replacing existing credentials with an attacker's credentials, modifying existing credentials, or obtaining the administrator or device credentials for use by the attacker.
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.
T.SECURITY_FUNCTIONALITY_FAILURE	A component of the network device may fail during start-up or during operations causing a compromise or failure in the security functionality of the network device, leaving the device susceptible to attackers.

3.3 Organizational Security Policies

The following table lists the Organizational Security Policies imposed by an organization to address its security needs.

Table 12 Organizational Security Policies

Policy Name	Policy Definition
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

This Chapter identifies the security objectives of the TOE and the IT Environment. The security objectives identify the responsibilities of the TOE and the TOE's IT environment in meeting the security needs.

- This document identifies objectives of the TOE as O.objective with objective specifying a unique name. Objectives that apply to the IT environment are designated as OE.objective with objective specifying a unique name.

4.1 Security Objectives for the TOE

The collaborative Protection Profile for Network Devices v1.0 does not define any security objectives for the TOE.

4.2 Security Objectives for the Environment

All of the assumptions stated in section 3.1 are considered to be security objectives for the environment. The following are the Protection Profile non-IT security objectives, which, in addition to those assumptions, are to be satisfied without imposing technical requirements on the TOE. That is, they will not require the implementation of functions in the TOE hardware and/or software. Thus, they will be satisfied largely through application of procedural or administrative measures.

Table 13 Security Objectives for the Environment

Environment Security Objective	IT Environment Security Objective Definition
OE.PHYSICAL	Physical security, commensurate with the value of the TOE and the data it contains, is provided by the environment.
OE.NO_GENERAL_PURPOSE	There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.
OE.NO_THRU_TRAFFIC_PROTECTION	The TOE does not provide any protection of traffic that traverses it. It is assumed that protection of this traffic will be covered by other security and assurance measures in the operational environment.
OE.TRUSTED_ADMIN	TOE Administrators are trusted to follow and apply all administrator guidance in a trusted manner.
OE.UPDATES	The TOE firmware and software is updated by an administrator on a regular basis in response to the release of product updates due to known vulnerabilities.
OE.ADMIN_CREDENTIALS_SECURE	The administrator's credentials (private key) used to access the TOE must be protected on any other platform on which they reside.

5 SECURITY REQUIREMENTS

This section identifies the Security Functional Requirements for the TOE. The Security Functional Requirements included in this section are derived from Part 2 of the *Common Criteria for Information Technology Security Evaluation, Version 3.1, Revision 4, dated: September 2012* and all international interpretations.

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: Indicated with **bold** text;
- Selection: Indicated with <u>underlined</u> text;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).

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

5.2 TOE Security Functional Requirements

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

Table 14 Security Functional Requirements

Class Name	Component Identification	Component Name
FAU: Security audit	FAU_GEN.1	Audit data generation
	FAU_GEN.2	User Identity Association
	FAU_STG_EXT.1	Protected Audit Event Storage
FCS: Cryptographic support	FCS_CKM.1	Cryptographic Key Generation (Refined)
	FCS_CKM.2	Cryptographic Key Establishment (Refined)
	FCS_CKM.4	Cryptographic Key Destruction
	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	HTTPS
	FCS_RBG_EXT.1	Cryptographic Operation (Random Bit Generation)
	FCS_SSHC_EXT.1	SSH Client Protocol
	FCS_SSHS_EXT.1	SSH Server Protocol
	FCS_TLSS_EXT.1	TLS Server Protocol
FIA: Identification and authentication	FIA_PMG_EXT.1	Password Management
	FIA_UIA_EXT.1	User Identification and Authentication
	FIA_UAU_EXT.2	Password-based Authentication Mechanism
	FIA_UAU.7	Protected Authentication Feedback
	FIA_X509_EXT.1	X.509 Certificate Validation
	FIA_X509_EXT.2	X.509 Certificate Authentication

Class Name	Component Identification	Component Name
	FIA_X509_EXT.3	X.509 Certificate Requests
FMT: Security management	FMT_MOF.1(1)/TrustedUpdate	TrustedUpdate Management of security functions behaviour
	FMT_MTD.1	Management of TSF Data
	FMT_SMF.1	Specification of Management Functions
	FMT_SMR.2	Restrictions on Security Roles
FPT: Protection of the TSF	FPT_SKP_EXT.1	Protection of TSF Data (for reading of all symmetric keys)
	FPT_APW_EXT.1	Protection of Administrator Passwords
	FPT_STM.1	Reliable Time Stamps
	FPT_TST_EXT.1	TSF Testing (Extended)
	FPT_TUD_EXT.1	Trusted Update
FTA: TOE Access	FTA_SSL_EXT.1	TSF-initiated Session Locking
	FTA_SSL.3	TSF-initiated Termination
	FTA_SSL.4	User-initiated Termination
	FTA_TAB.1	Default TOE Access Banners
FTP: Trusted Path/Channels	FTP_ITC.1	Inter-TSF trusted channel
	FTP_TRP.1	Trusted Path

5.3 SFRs Drawn from NDcPP ONLY

5.3.1 Security audit (FAU)

5.3.1.1 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 <u>not specified</u> 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).*
- *Security related 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).*
- *Starting and stopping services (if applicable)*
- <u>no other actions</u>;

d) *Specifically defined auditable events listed in* Table 15.

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 PP/ST, *information specified in column three of* Table 15.

Table 15 Auditable Events

SFR	Auditable Event	Additional Audit Record Contents
Audit Events and Details from NDcPP		
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(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 an HTTPS session.	Reason for failure.
FCS_RBG_EXT.1	None.	None.
FCS_SSHC_EXT.1	Failure to establish an SSH session	Reason for failure.
	Successful SSH rekey	Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure.
	Successful SSH rekey	Non-TOE endpoint of connection (IP address) for both successes and failures.
FCS_TLSS_EXT.1	Failure to establish an TLS session	Reason for failure.
FIA_PMG_EXT.1	None.	None.
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt (e.g., IP address).
FIA_UAU_EXT.2	All use of the authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	None.
FMT_MOF.1(1)/Trusted Update	Any attempt to initiate a manual update	None.
FIA_X509_EXT.1	Unsuccessful attempt to validate a certificate	Reason for failure
FIA_X509_EXT.2	None.	None.
FIA_X509_EXT.3	None.	None.
FMT_MTD.1	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_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; result of the update attempt (success and failure)	No additional information.
FTA_SSL_EXT.1	Any attempts at unlocking of an interactive session.	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.

SFR	Auditable Event	Additional Audit Record Contents
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 path. Termination of the trusted path. Failures of the trusted path functions.	Identification of the claimed user identity.

5.3.1.2 FAU_GEN.2 User Identity Association

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

5.3.1.3 FAU_STG_EXT.1 Protected Audit Event Storage

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

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

FAU_STG_EXT.1.3 The TSF shall [overwrite previous audit records according to the following rule: *when allotted space has reached its threshold, no other action*] when the local storage space for audit data is full.

5.3.2 Cryptographic Support (FCS)

5.3.2.1 FCS_CKM.1 Cryptographic Key Generation (Refined)

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

- **RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;**
- **ECC schemes using "NIST curves" P-256, P-384, P-521 that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;**

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

5.3.2.2 FCS_CKM.2 Cryptographic Key Establishment (Refined)

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

- **RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B, "Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography";**

- **Elliptic curve-based key establishment schemes that meets the following: NIST Special Publication 800-56A, “Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography”;**

]

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

5.3.2.3 FCS_CKM.4 Cryptographic Key Destruction

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

For plaintext keys in volatile storage, the destruction shall be executed by a [single overwrite *consisting of* [zeroes]]*;*

• _*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* [

- o logically addresses the storage location of the key and performs a [single-pass] overwrite consisting of [zeroes]*;*

]
that meets the following: No Standard.[3]

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

FCS_COP.1.1(1) Refinement: The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in* [CBC] mode and cryptographic key sizes [128 bits, 256-bits] that met the following: *AES as specified in ISO 18033-3*, [CBC as specified in ISO 10116].

5.3.2.5 FCS_COP.1(2) Cryptographic Operation (for cryptographic signature)

FCS_COP.1.1(2) Refinement: 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 or greater*]

that meets 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 RSASSAPKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3] [4]

[3] TD0130

[4] TD0116

5.3.2.6 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, SHA-384, SHA-512] ~~and cryptographic key sizes~~ ~~[assignment: cryptographic key sizes]~~ that meet the following: ISO/IEC 10118-3:2004.

5.3.2.7 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] and cryptographic key sizes [*160 key size (in bits) used in HMAC*] **and message digest sizes** [**160**] **bits** that meet the following: ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2".

5.3.2.8 FCS_HTTPS_EXT.1 Explicit: 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 HTTPS using TLS in accordance with [FCS_TLSS_EXT.1].

FCS_HTTPS_EXT.1.3 The TSF shall establish the connection only if [the peer initiates handshake].[5]

5.3.2.9 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 ISO/IEC 18031:2011 using [Hash_DRBG (any), HMAC_DRBG (any), CTR_DRBG (AES)].

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

5.3.2.10 FCS_SSHC_EXT.1 Explicit: SSH

FCS_SSHC_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [no other RFCs].

FCS_SSHC_EXT.1.2 The TSF shall ensure that the SSH protocol implementation supports the following authentication methods as described in RFC 4252: public key-based, [no other method].

[5] TD0125

FCS_SSHC_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*35,000*] bytes in an SSH transport connection are dropped.

FCS_SSHC_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: aes128-cbc, aes256-cbc, [no other algorithms].

FCS_SSHC_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses [ssh-rsa] and [no other public key algorithms] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHC_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [hmac-sha1] and [no other MAC algorithms] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHC_EXT.1.7 The TSF shall ensure that [diffie-hellman-group14-sha1] and [no other methods] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHC_EXT.1.8 The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

FCS_SSHC_EXT.1.9 The TSF shall ensure that the SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key or [no other methods] as described in RFC 4251 section 4.1.

5.3.2.11 FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and [no other RFCs].

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

FCS_SSHS_EXT.1.3 The TSF shall ensure that, as described in RFC 4253, packets greater than [*35,000*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms: aes128-cbc, aes256-cbc, [no other algorithms].

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH transport implementation uses [ssh-rsa] and [no other methods] as its public key algorithm(s) and rejects all other public key algorithms.

FCS_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [hmac-sha1] and [no other MAC algorithms] as its data integrity MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [diffie-hellman-group14-sha1] and [no other methods] are the only allowed key exchange methods used for the SSH protocol.

FCS_SSHS_EXT.1.8 The TSF shall ensure that within SSH connections the same session keys are used for a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. After either of the thresholds are reached a rekey needs to be performed.

5.3.2.12 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)] supporting the following ciphersuites:

- *Mandatory Ciphersuites:*
  - *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
- *[Optional Ciphersuites:*
  - *TLS_RSA_WITH_AES_256_CBC_SHA as defined in RFC 3268*
  - *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_RSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5246*
  - *TLS_RSA_WITH_AES_256_CBC_ SHA256 as defined in RFC 5246*
  - *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*
  - *no other ciphersuite*].

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 key establishment parameters using RSA with key size 2048 bits and [no other size] and [no other].

5.3.3 Identification and authentication (FIA)

5.3.3.1 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;

5.3.3.2 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 action on behalf of that administrative user.

5.3.3.3 FIA_UAU_EXT.2 Extended: Password-based Authentication Mechanism

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

5.3.3.4 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.

5.3.3.5 FIA_X509_EXT.1 X.509 Certificate Validation

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

- RFC 5280 certificate validation and certificate path validation.
- The certificate path must terminate with a trusted CA certificate.
- The TSF shall validate a certificate path by ensuring the presence of the basicConstraints extension and that the CA flag is set to TRUE for all CA certificates.
- The TSF shall validate the revocation status of the certificate using [a Certificate Revocation List (CRL) as specified in RFC 5759].
- The TSF shall validate the extendedKeyUsage field according to the following rules:

- o 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.
- o 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.
- o *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.*

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

5.3.3.6 FIA_X509_EXT.2 X.509 Certificate Authentication

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

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

5.3.3.7 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.

5.3.4 Security management (FMT)

5.3.4.1 FMT_MOF.1(1)/TrustedUpdate Management of security functions behaviour

FMT_MOF.1.1(1)/TrustedUpdate The TSF shall restrict the ability to enable the functions *to perform manual update* to *Security Administrators*.

5.3.4.2 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*.

5.3.4.3 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* [hash comparison] *capability prior to installing those updates;*
- [Ability to configure audit behavior;
- Ability to configure the cryptographic functionality.]

5.3.4.4 FMT_SMR.2 Restrictions on Security Roles

FMT_SMR.2.1 The TSF shall maintain the roles:

- *Security Administrator.*

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

- *The Security Administrator role shall be able to administer the TOE locally;*
- *The Security Administrator role shall be able to administer the TOE remotely* are satisfied.

5.3.5 Protection of the TSF (FPT)

5.3.5.1 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.

5.3.5.2 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.

5.3.5.3 FPT_STM.1 Reliable time stamps

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

5.3.5.4 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: *FIPS Self-Tests*.

5.3.5.5 FPT_TUD_EXT.1 Extended: 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 as well as the most recently installed version of the TOE firmware/software.

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

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

5.3.6 TOE Access (FTA)

5.3.6.1 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.

5.3.6.2 FTA_SSL.3 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*.

5.3.6.3 FTA_SSL.4 User-initiated Termination

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

5.3.6.4 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.

5.3.7 Trusted Path/Channels (FTP)

5.3.7.1 FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 Refinement: The TSF shall **be capable of using** [SSH] 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 [*communications with the following*:

- *external audit servers using SSH*].

5.3.7.2 FTP_TRP.1 Trusted Path

FTP_TRP.1.1 Refinement: The TSF shall **be capable of using SSH and 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 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.**

5.4 TOE SFR Dependencies Rationale for SFRs Found in NDcPP

The NDcPPv1.0 contains all the requirements claimed in this Security Target. As such the dependencies are not applicable since the PP itself has been approved.

5.5 Security Assurance Requirements

5.5.1 SAR Requirements

The TOE assurance requirements for this ST are taken directly from the NDcPP which are derived from Common Criteria Version 3.1, Revision 4. The assurance requirements are summarized in the table below.

Table 16: Assurance Measures

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)	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative User guidance
Life cycle support (ALC)	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM coverage
Tests (ATE)	ATE_IND.1	Independent testing - sample
Vulnerability assessment (AVA)	AVA_VAN.1	Vulnerability survey

5.5.2 Security Assurance Requirements Rationale

The Security Assurance Requirements (SARs) in this Security Target represent the SARs identified in the NDcPPv1.0. As such, the NDcPP SAR rationale is deemed acceptable since the PP itself has been validated.

5.6 Assurance Measures

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

Table 17 Assurance Measures

Component	How requirement will be met
ADV_FSP.1	There are no specific assurance activities associated with ADV_FSP.1. The requirements on the content of the functional specification information is implicitly assessed by virtue of the other assurance activities being performed. The functional specification is comprised of the information contained in the AGD_OPE and AGD_PRE documentation, coupled with the information provided in the TSS of the ST. The assurance activities in the functional requirements point to evidence that should exist in the documentation and TSS section; since these are directly associated with the SFRs, the tracing in element ADV_FSP.1.2D is implicitly already done and no additional documentation is necessary.

Component	How requirement will be met
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 AGD and ST implicitly meet this assurance requirement. The evaluator shall check the ST to ensure that it contains an identifier (such as a product name/version number) that specifically identifies the version that meets the requirements of the ST. Further, the evaluator shall check the AGD guidance and TOE samples received for testing to ensure that the version number is consistent with that in the ST.
ALC_CMS.1	
ATE_IND.1	Cisco will provide the TOE for testing.
AVA_VAN.1	Cisco will provide the TOE for testing.

6 TOE SUMMARY SPECIFICATION

6.1 TOE Security Functional Requirement Measures

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

Table 18 How TOE SFRs are Met

TOE SFRs	How the SFR is Met
FAU_GEN.1	The TOE generates an audit record that is stored internally within the TOE whenever an audited event occurs. Audit records are stored in files in the file system provided by the TOE's modified BSD operating system component. The TOE stores auditable events in separate log files containing related types of audited data. The following log files together comprise the TSF audit trail by covering all events listed in Table 16: • IronPort Text Mail Logs -record information regarding the operations of the email system, such as message receipt, message delivery attempts, bounces, etc. • Delivery Logs -record critical information about the TOE's email delivery operations • Bounce Logs - record information about bounced recipients • System Logs - record boot information and DNS status information • CLI Audit Logs - record all CLI activity • HTTP Logs (GUI Audit Logs) - HTTP logs record information about the secure HTTP services enabled on the interface. Because the graphical user interface (GUI) is accessed via HTTPS, the HTTP logs are ostensibly the GUI equivalent of the CLI Audit logs. Session data (new session, session expired) and pages accessed in the GUI are recorded. • Anti-Virus Logs - record events related to the status of receiving updates of the latest anti-virus identity files • Authentication Logs - record all successful user logins and failed user authentication attempts. • NTP Logs – record events regarding NTP time changes • Update Logs – record events for TOE updates Note that the TOE generates various other log files that record information about the behavior of the TOE, but these do not contain logs that satisfy the TOE's auditing requirements. The TOE shall ensure that each auditable event is associated with the user that triggered the event and as a result they are traceable to a specific user. 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 presented. Each audit record includes date and time of the audited event, type of event, subject identity, and the outcome (success or failure) of the event. The auditable events comprise: • Start-up and shutdown of the audit function - recorded in System Logs

TOE SFRs	How the SFR is Met
	• Access to System - recorded in IronPort Text Mail Logs, Delivery Logs and Bounce Logs • Access to the TOE and System data - recorded in: IronPort Text Mail Logs, Delivery Logs, and Bounce Logs (for email traffic); CLI Audit Logs (for console interfaces) and HTTP logs for GUI; and Update logs (TOE updates) • Reading of information from the audit records - recorded in CLI Audit Logs and HTTP logs for GUI • Unsuccessful attempts to read information from the audit records - recorded in CLI Audit Logs and HTTP logs for GUI • All modifications to the audit configuration that occur while the audit collection functions are operating - recorded in CLI Audit Logs and HTTP logs for GUI • All use of the authentication mechanism - recorded in Authentication Logs • All use of the user identification mechanism - recorded in Authentication Logs • All modifications in the behavior of the functions of the TSF - recorded in CLI Audit Logs and HTTP logs for GUI • All modifications to the values of TSF data - recorded in CLI Audit Logs, HTTP logs for GUI, and NTP Logs • Modifications to the group of users that are part of a role - recorded in CLI Audit Logs and HTTP logs for GUI • The action taken upon detection of modification of transmitted TSF data -recorded in Anti-Virus Logs. Administrators and Operators can access all audit information. The administrators can manually download the log files by clicking a link to the log directory on the Log Subscriptions page, then clicking the log file to access. Depending on the browser, an authorized administrator can view the file in a browser window, or open or save it as a text file. This method uses the HTTP(S) protocol and is the default retrieval method. Example audit events are included below: < Date and time of the event> < type of event> < source IP> <subject identity> <outcome> <url accessed with return HTTP headers> Thu Nov 1 19:03:00 2012 Info: login:10.65.79.90 user:admin session:XtL50wP9GB92YfjVerYb Thu Nov 1 19:03:00 2012 Info: req:10.65.79.90 user:- id:XtL50wP9GB92YfjVerYb 303 POST /login HTTP/1.1 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/537.4 (KHTML, like Gecko) Chrome/22.0.1229.94 Safari/537.4 Thu Nov 1 19:03:02 2012 Info: req:10.65.79.90 user:admin id:XtL50wP9GB92YfjVerYb 200 GET /monitor/user_report HTTP/1.1 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/537.4 (KHTML, like Gecko) Chrome/22.0.1229.94 Safari/537.4 Thu Nov 1 19:03:03 2012 Info: req:10.65.79.90 user:admin id:XtL50wP9GB92YfjVerYb 200 GET /scfw/1y-8.0.0-366/navigation.css HTTP/1.1 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/537.4 (KHTML, like Gecko) Chrome/22.0.1229.94 Safari/537.4 Thu Nov 1 19:03:03 2012 Info: req:10.65.79.90 user:admin

TOE SFRs	How the SFR is Met
	id:XtL50wP9GB92YfjVerYb 200 GET /scfw/1y-8.0.0-366/widget/tablecols/table-cols-min.css HTTP/1.1 Mozilla/5.0(Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/537.4 (KHTML, like Gecko) Chrome/22.0.1229.94 Safari/537.4 Thu Nov 1 19:03:03 2012 Info: req:10.65.79.90 user:admin id:XtL50wP9GB92YfjVerYb 200 GET /yui_webui HTTP/1.1 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/537.4 (KHTML, like Gecko) Chrome/22.0.1229.94 Safari/537.4 Thu Nov 1 19:03:04 2012 Info: req:10.65.79.90 user:admin id:XtL50wP9GB92YfjVerYb 200 GET /javascript?CSRFKey=f0fadf9c-fce3-43b6-84ae-3b42f559bcd5&language=en-usHTTP/1.1 Mozilla/5.0 (Macintosh; Intel Mac OS X 10_7_4) AppleWebKit/537.4 (KHTML, like Gecko) Chrome/22.0.1229.94 Safari/537.4
FAU_GEN.2	The TOE shall ensure that each auditable event is associated with the user that triggered the event and as a result they are traceable to a specific user. 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 presented. A sample audit record is below: Fri Jun 6 16:35:22 2014 Info: login:192.168.1.228 user:admin session:fI023Y0gCdc5u4BuWqL8 The HTTPS session has been established successfully.
FAU_STG_EXT.1	The TOE is configured to export the audit log records within each of the log files listed below to a specified, external syslog server. The TOE protects communications with an external syslog server via SCP over SSH. This must be configured by an authorized administrator. Once configured, the TOE can automatically provide the audit backups. The log files that must be configured for export are: • IronPort Text Mail Logs • Delivery Logs • Bounce Logs • System Logs • CLI Audit Logs • HTTP Logs (GUI Audit Logs) • Anti-Virus Logs • Authentication Logs Note that the TOE can also export various other log file's audit records to an external syslog server, but these other log files do not contain logs that satisfy the TOE's auditing requirements. The TOE provides the following mechanisms for retrieving log files: • SCP - a CLI client that supports an scp command can copy log files from the TOE to the client host. The user of the scp command on the client must be the admin user on the TOE, as the TOE will prompt for the admin user password before processing the SCP request • SCP Push - additionally, the TOE can be configured to periodically push log files to a SCP server on a remote computer Both of the above SCP methods are secured by SSHv2. The SCP is the method that periodically pushes log files to an SCP server on a remote syslog

TOE SFRs	How the SFR is Met
	server. This method requires an SSH SCP server on a remote computer using SSHv2 protocol. The subscription requires a username, SSH key, and destination directory on the remote syslog server. Log files are transferred based on a rollover schedule set by the authorized administrator. The TOE generates an email alert to the authorized administrator or System administrator and begins overwriting the oldest stored audit records when the audit trail becomes full. (Note that the TOE does not stop collecting or producing System data). The alert is generated to an authorized administrator or System administrator who has been configured via the Command Line Interface (`alertconfig` command) to receive email alerts for this event. The TOE does not provide interfaces to modify individual records. When the audit trail becomes full, the TOE ensures that the most recent audit records will be maintained, limited only by the available storage space. This method periodically pushes log files to an SCP server on a remote syslog server. This method requires an SSH SCP server on a remote computer using SSHv2 protocol. The subscription requires a username, SSH key, and destination directory on the remote syslog server. Log files are transferred based on a rollover schedule set by the authorized administrator. The TOE is capable of detecting when the SSH connection fails. The TOE also stores a local set of audit records on the TOE, and continues to do so if the communication with the syslog server goes down. The TOE stores the audit logs locally as configured with the logconfig command in the CLI and the Log Subscriptions page in the GUI. The size of the local log files are set by an authorized administrator using the 'Rollover by File Size' configuration setting. Once the file reaches the specified size it is sent to the syslog server using SCP. These transfers can also be configured based on configured time intervals. If the SSH connection fails, the log files will remain on the TOE. On the next SCP push based on either the maximum log file size being exceeded or on the time interval, the current log file and the log files previously unsuccessfully transferred will be transferred. Only Authorized Administrators are able to clear the local logs, and there is no TOE interface that allows for administrators to modify the contents of the local audit records. The TOE's default installation configures the audit log files to maintain 10 files of no more than 10MB for each log subscription. The administrative user does not need to configure this. However, this value is customizable. The administrative user can configure each log subscription to allow 1-1000 maximum log files, and each log file can be configurable to a maximum of between 100KB and 100MB. There is no limit to the number of log subscriptions that the administrative user can create. With a typical configuration, the log space should not grow beyond a reasonable limit. If through customization of the log limits, the log files grow too much, alerts will be sent to the administrator when the log partition grows beyond 90% usage. If the space available for storing audit records is exhausted, the TOE will start to overwrite the oldest records in the audit trail, and generate an email alert to this effect and send it to an Administrator.
FCS_CKM.1 FCS_CKM.2	The TOE generates asymmetric keys in accordance with the RSA schemes using key sizes of 2048-bit or greater that are conformant to the NIST SP 800-56B. In addition, ECC schemes are used with P-256, P-384, and P-521. The

TOE SFRs	How the SFR is Met
	TOE can create a RSA public-private key pair that can be used to generate a Certificate Signing Request (CSR). Via offline CSR the TOE can send the CSR to a Certificate Authority (CA) for the CA to generate a certificate; and receive its certificate (including X.509v3) from the CA. Integrity of the CSR and certificate during transit are assured through use of digital signatures (encrypting the hash of the TOE's public key contained in the CSR and certificate). The TOE can store and distribute the certificate to external entities including Registration Authorities (RA). The TOE can also use X.509v3 certificates for authentication of TLS sessions. The TOE acts as both a sender and receiver for RSA-based key establishment schemes 800-56A and 800-56B. The key pair generation portions of "The RSA Validation System" for FIPS 186-4 were used as a guide in testing the FCS_CKM.1 during the FIPS validation. For details on each protocol see the related SFR. ECDH is used for key exchange for the SSHv2 sessions.
FCS_CKM.4	The TOE meets all requirements specified in FIPS 140-2 for destruction of keys and Critical Security Parameters (CSPs) in that none of the symmetric keys, pre-shared keys, or private keys are stored in plaintext form as described in Table 19. Through the implementation of the cryptographic module, the TOE zeroizes all of the cryptographic keys used within the TOE after the key is no longer of use to the TOE. The cryptographic module performs the overwrite of the cryptographic keys and other critical security parameters that are handled by the CiscoSSL library are zeroized using a function that will overwrite the memory with random data once they are no longer in use. Swap space is encrypted using AES to avoid accidental leakage of CSPs. As part of the reload command, an option to wipe the data is provided. The wipe option along with the 'wipedata' command will overwrite the hard drive with zeros so that the keys are zeroized within the old core dump files. See Table 20 in Section 7.1, below for more information. The key and CSP zeroization capabilities of the TOE's cryptographic module have been verified as part of the TOE's FIPS 140-2 validation #2505.
FCS_COP.1(1)	The TOE provides symmetric encryption and decryption capabilities using AES in CBC mode (128 and 256 bits) as described in ISO 18033-3 and ISO 10116. AES is implemented in the following protocols: TLSv1.1, v1.2, and SSHv2. The TOE provides AES encryption and decryption in support of SSHv2 and TLSv1.2 for secure communications. Management of the cryptographic algorithms is provided through the CLI with auditing of those commands.
FCS_COP.1(2)	The TOE provides cryptographic signature services using the following: • RSA Digital Signature Algorithm with key size of 2048 and greater as specified in FIPS PUB 186-4, "Digital Signature Standard". The TOE provides cryptographic signatures in support of SSHv2, and TLS for secure communications. Management of the cryptographic algorithms is provided through the CLI with auditing of those commands. The TOE provides the RSA option in support of SSHv2 and TLS key establishment.

TOE SFRs	How the SFR is Met
	RSA (3072-bit and 4096-bit) is used in the establishment of SSHv2 key establishment. For SSHv2, RSA host keys are supported
FCS_COP.1(3)	The TOE provides cryptographic hashing services using SHA-1, SHA-256, SHA-384, and SHA-512 as specified in ISO/IEC 10118-3:2004. The TOE provides hashing as part of the TLS session integrity. ESA uses server side X.509v3 certificates for authentication. Digital signature is comprised of implementing an encrypted hash function. Verification of the digital signature includes the process of decrypting the encrypted hash and verifying the hash is valid. SHA1 is also used in the keyed hash function of HMAC. The TOE provides Secure Hash Standard (SHS) hashing in support of TLS, for secure communications. Management of the cryptographic algorithms is provided through the CLI with auditing of those commands.
FCS_COP.1(4)	The TOE provides keyed-hashing message authentication services using HMAC-SHA1, key size 160 bits, and message digest sizes 160 bits as specified in ISO/IEC 9797-2:2011, Section 7 “MAC Algorithm 2”. The block size for HMAC-SHA1 is 512 bits. The TOE provides SHS hashing and HMAC message authentication in support of SSHv2, and TLSv1.2 for secure communications. Management of the cryptographic algorithms is provided through the CLI with auditing of those commands. SHS hashing and HMAC message authentication (SHA-1) is used in the establishment of TLS/HTTPS, and SSHv2 sessions.
FCS_HTTPS_EXT.1	The TOE implements HTTPS over TLS as specified in RFC 2818 and FCS_TLSS_EXT.1. The TSF HTTPS implementation authenticates the TOE to the remote client with an X.509 certificate. System Administrators manage the TOE identity certificates using the Destination Controls page in the GUI or destconfig command in the TOE CLI. HTTPS uses the Security Administrator - selected identity certificate. The TSF HTTPS implementation performs server based authentication using a server X.509v3 certificate to establish the TLS session. The TSF HTTPS implementation does not require client authentication at the TLS level but presents the Web interface logon page for administrative users to authenticate using their name and password.
FCS_RBG_EXT.1	The TOE implements a NIST-approved Deterministic Random Bit Generator (DRBG), as specified in ISO/IEC 18031:2011 per NIST SP800-90A using Hash_DRBG (any), HMAC_DRBG (any), and CTR_DRBG (AES). The TOE models provide a software based entropy source as described in FCS_RBG_EXT.1. This output is used directly to seed the DRBG. The TOE implements a random number generator for Elliptic Curve Diffie-Hellman based key establishment (conformant to NIST SP 800-56A) and for RSA key establishment schemes (conformant to NIST SP 800-56B). The TOE complies with section 5.6 and all subsections regarding asymmetric key pair generation and key establishment in the NIST SP 800-56A. The TOE complies with section 6 and all subsections regarding RSA key pair generation and key establishment in the NIST SP 800-56B. The TOE includes a FIPS 140-2 Level 1 validated cryptomodule, Cisco FIPS Object Module (FOM) (Software Version: 6.0) cert #2505.

TOE SFRs	How the SFR is Met
	The TOE is able to generate asymmetric key pairs with modulus 2048 bits which is equivalent to a symmetric key strength of 112 bits.
FCS_SSHC_EXT.1	The TOE implements SSHv2 to secure the remote session between the TOE and syslog server. SSHv2 is implemented according to the following RFCs: 4251, 4252, 4253, and 4254. The TOE supports public key-based authentication. The TOE uses a SCP push to securely send the audit logs to a remote syslog server over a secured SSHv2 session. The SSH client authenticates the identity of the SSH server using a local database associating each host name with its corresponding public key RFC 4251 section 4.1. SSH connections will be dropped if the TOE receives a packet larger than 35,000 bytes. Large packets are detected by the SSH implementation, and dropped internal to the SSH process. A rekey occurs after a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. The key exchange methods used by the TOE is a configurable option but diffie-hellman-group14-sha1 is the only allowed methods within the evaluated configuration. Any session where the SSH server offers only non compliant algorithms or key sizes per the NDcPP will be rejected by the SSH client. SSH sessions can only be established when compliant algorithms and key sizes can be negotiated. The TOE implementation of SSHv2 supports the following: • public key algorithms for authentication: ssh-rsa. • encryption algorithms, aes128-cbc, aes256-cbc to ensure confidentiality of the session. • hashing algorithms HMAC-SHA1 to ensure the integrity of the session.
FCS_SSHS_EXT.1	The TOE implements SSHv2 for remote CLI sessions (telnet is disabled in the evaluated configuration). SSHv2 is implemented according to the following RFCs: 4251, 4252, 4253, and 4254. The TOE supports both public key-based and password-based authentication. Remote CLI SSHv2 sessions are limited to an administrator configurable session timeout period, and will be rekeyed after a threshold of no longer than one hour, and no more than one gigabyte of transmitted data. SSHv2 connections will be dropped if the TOE receives a packet larger than 35,000 bytes. Large packets are detected by the SSH implementation, and dropped internal to the SSH process. The key exchange methods used by the TOE is a configurable option but diffie-hellman-group14-sha1 is the only allowed methods within the evaluated configuration. Any session where the SSH client offers only non compliant algorithms or key sizes per the NDcPP will be rejected by the SSH server. SSH sessions can only be established when compliant algorithms and key sizes can be negotiated. The TOE implementation of SSHv2 supports the following: • public key algorithms for authentication: ssh-rsa. • password-based authentication for administrative users accessing the TOE's CLI through SSHv2. • encryption algorithms, aes128-cbc, aes256-cbc to ensure confidentiality of the session. • hashing algorithms HMAC-SHA1 to ensure the integrity of the session.
FCS_TLSS_EXT.1	An authorized administrator can initiate inbound TLSv1.1 and TLSv1.2 connections using the web based GUI for remote administration of the TOE.

TOE SFRs	How the SFR is Met
	Any session where the client offers the following in the client hello: SSL 2.0, SSL 3.0 and TLS 1.0 will be rejected by the TLS server. Since RSA is being used for key exchange and authentication there are no specific parameters associated with the server key exchange. The TOE uses TLS for inbound/outbound email handling. Using the below TLS_RSA ciphers the RSA public key is used for authentication and key exchange. Using the below TLS_DHE ciphers the standard diffie hellman paramaters P, Q, and G are used for key exchange. TLS is also used to protect the TLS sessions with the TOE, which supports the mandatory ciphersuite as well as the following optional ciphersuite: • TLS_RSA_WITH_AES_128_CBC_SHA • TLS_RSA_WITH_AES_256_CBC_SHA • TLS_DHE_RSA_WITH_AES_128_CBC_ • 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
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: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(”, and “)”. Minimum password length is settable by the Authorized Administrator, and can be configured for minimum password lengths of 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, except for the login banner that is displayed prior to user authentication. Administrative access to the TOE is facilitated through the TOE’s CLI and web based GUI. The TOE mediates all administrative actions through the CLI and web based GUI. Once a potential administrative user attempts to access the CLI via either a directly connected console or remotely through SSHv2, the TOE prompts the user for a user name and password. Likewise, when a potential administrative user attempts to access the web based GUI of the TOE through a TLSv1.1 or 1.2/HTTPS, the TOE prompts the user for a user name and password. Only after the administrative user presents the correct authentication credentials will access to the TOE administrative functionality be granted. No access is allowed to the administrative functionality of the TOE until an administrator is successfully identified and authenticated. The TOE provides a local password based authentication mechanism for the CLI when accessed both locally and remotely as well as the GUI. When the CLI is accessed remotely, the session is secured via SSHv2 and authenticated using SSH public key. The password mechanism can be configured to require passwords to be a minimum of 15 characters from the printable character set. The TOE prevents administrative user actions from being performed prior to identification and authentication of the user (all filtering of email occurs without identification or authentication of users). The TOE defines a default user account, called `admin`. This account has all administrative privileges. The TOE allows additional administrative accounts to be created. Each account comprises a user name (which identifies the user), authentication data, in the form of a password, and authorizations, in the

TOE SFRs	How the SFR is Met
	form of a group assignment that grants certain administrative privileges. Assigning a group to a user account essentially confers a security management role on that user. Note, however, that users accessing the CLI via SSHv2 can be authenticated using public key cryptography. This requires the user's public key to be entered into the TOE (using the sshconfig command) and associated with the user's account. If there is no public key configured for the user, the user will instead be prompted to enter a password to authenticate.
FIA_UAU.7	When a user enters their password at the local console, the TOE does not echo any characters as they are entered as such the user password is obscured in both the CLI and GUI. Likewise, when for the remote CLI and GUI session authentication, the TOE does not echo any characters as they are entered.
FIA_X509_EXT.1 FIA_X509_EXT.2 FIA_X509_EXT.3	The TOE uses X.509v3 certificates as defined by RFC 5280 to support authentication for TLS connections. The certificate validation checking takes place during the TLS session setup. The TOE supports the following methods to obtain a certificate from a CA: • Manual cut-and-paste - ESA generates the Certificate Request Message as described in RFC 2986 which contains the public key and is displayed via the GUI or CLI interface. This allows the administrator to copy the certificate request and in a secure offline manner send the request to a Certification Authority to be transformed into an X.509v3 public-key certificate. • Both the certificate request message and the certificates themselves provide protection in that they are digitally signed. If a certificate is modified in any way, it would be invalidated. The digital signature verifications process would show that the certificate had been tampered with when the hash value would be invalid. • The certificate chain establishes a sequence of trusted certificates, from a peer certificate to the root CA certificate. Within the PKI hierarchy, all enrolled peers can validate the certificate of one another if the peers share a trusted root CA certificate or a common subordinate CA. When a certificate chain is received from a peer, the default processing of a certificate chain path continues until the first trusted certificate is reached. • The authorized administrator can also configure one or more certificate fields as listed below that will be used to compare the imported certificate to specific criteria such as: • alt-subject-name (If subject name doesn't match request, then the alternative subject name filed is used) • expires-on (If certificate is expired, rejects certificate) • issuer-name (Is there a trusted root certificate installed for the CA that signed the certificate). • name (Does the name in the request match the name in the certificate) • serial-number (Has the certificate been revoked. Serial number will be in the CRL/OCSP) • subject-name (Does the name in the request match the name in the certificate) More than one certificate from one or more CAs on the TOE can be stored and used by ESA. For example, one certificate from one CA could be used

TOE SFRs	How the SFR is Met
	for SSH connections, while another certificate from another CA could be used for TLS connections. The administrative user manually installs and selects the certificate used by the TOE for each certificate. The physical security of the TOE (A.PHYSICAL_PROTECTION) protects the switch and the certificates from being tampered with or deleted. In addition, the TOE identification and authentication security functions protect an unauthorized user from gaining access to the TOE. The use of CRL is configurable and may be used for certificate revocation. • crl --Certificate checking is performed by a CRL. This is the default option. Checking is also done for the basicConstraints extension and the CA flag to determine whether they are present and set to TRUE. The local certificate that was imported must contain the basic constraints extension with the CA flag set to true, the check also ensure that the key usage extension is present, and the keyEncipherment bit or the keyAgreement bit or both are set. If they are not, the certificate is not accepted. If the connection to determine the certificate validity cannot be established, ESA does not accept the certificate.
FMT_MOF.1.1(1)/TrustedUpdate	Manual software updates can only be done by the authorized administrator through either the CLI or GUI. These updates include software upgrades.
FMT_MTD.1	The TOE provides administrative users with a CLI and web based GUI to interact with and manage the security functions of the TOE. The CLI is the main interface used to administer the TOE since all functionality to configure and monitor the system is here. The GUI contains most of the functionality an authorized administrator needs to configure and monitor the system. However, not all CLI commands are available in the GUI; some features are *only* available through the CLI. The CLI is used to perform all security functions, including configuring the ESA appliance and managing users and email security policies. No administrative functionality is available prior to administrative login. Through the CLI, the TOE provides the ability for Authorized Administrators to manage TOE data, such as audit data, configuration data, security attributes, message filters, login banners, and mail policies via the CLI and GUI. A subset of functionality is available in the GUI. Each of the predefined and administratively configured privilege levels has a set of permissions that will grant them access to the TOE data, though with some privilege levels, the access is limited. The TOE performs role-based authorization, using TOE platform authorization mechanisms, to grant access to the semi-privileged and privileged levels. The term "Authorized Administrator" is used in this ST to refer to any user which has been assigned to a privilege level that is permitted to perform the relevant action; therefore has the appropriate privileges to perform the requested functions. Therefore, semi-privileged administrators with only a subset of privileges can also modify TOE data based on if granted the privilege. See FMT_SMR.2 for more details on the TOE roles and related privileges.
FMT_SMF.1	The TOE provides all the capabilities necessary to securely manage the TOE. The administrative user can connect to the TOE using the CLI to perform

TOE SFRs	How the SFR is Met
	these functions via SSHv2, the local console, or via the GUI over TLS/HTTPS. The specific management capabilities available from the TOE include: • Local and remote administration of the TOE and the services provided by the TOE via the TOE CLI, as described above; • The ability to update the AsyncOS software (image integrity verification is provided using SHA-384) • Ability to configure the cryptographic functionality; • Ability to enable, disable, determine and modify the behavior of all the security functions of the TOE via the CLI and GUI. • Configure access banner • Configure session inactivity time before session termination or locking, • Ability to configure the audit behavior.
FMT_SMR.2	The TOE platform maintains both privileged and semi-privileged administrator roles. The terms "Authorized Administrator" and "Security Administrator" are used interchangeable in this ST to refer to any user that has been assigned to a privilege level that is permitted to perform the relevant action; therefore, has the appropriate privileges to perform the requested functions. The assigned role determines the functions the user can perform; hence the authorized administrator with the appropriate privileges. The TOE performs role-based authorization, using TOE platform authorization mechanisms, to grant access to the semi-privileged and privileged roles. The default user account for the system, admin, has all administrative privileges. The admin user account cannot be deleted, but an authorized administrator can change the password and lock the account. When an authorized administrator creates a new user account, they can assign the user to a predefined or a custom user role. Each role contains differing levels of permissions within the system. Although there is no limit to the number of user accounts that an authorized administrator can create on the appliance, authorized administrator cannot create user accounts with names that are reserved by the system such as "operator" or "root." The following roles are predefined by the system and can be assigned to user accounts: • admin - default user account that has full access to all system configuration settings. • Administrator - has full access to all system configuration settings. • Technician - can perform system upgrades, reboot the appliance, and manage key features. • Operators - are restricted from creating, editing, or removing user accounts and cannot use the following commands: resetconfig, upgradecheck, upgradeinstall, systemsetup or running the System Setup Wizard. • Guest - can only view system status information.

TOE SFRs	How the SFR is Met
	• Read-Only Operator - can view administrative interfaces, but do not have the ability to commit configuration changes or to access the file system or SCP, thus preventing them from accessing log files • Help Desk User - have access to system quarantines, end-user spam quarantines, and message tracking via the GUI. • Custom user role - can only access email security features assigned to the role. These features can be any combination of DLP policies, email policies, reports, quarantines, local message tracking, encryption profiles, and the Trace debugging tool. The users cannot access system configuration features. Only administrators can define custom user roles. The TOE can and shall be configured to authenticate all access to the CLI and GUI using a username and password. The TOE supports both local administration via a directly connected console cable and remote authentication via SSHv2 and TLS/HTTPS.
FPT_SKP_EXT.1 FPT_APW_EXT.1	The TOE stores all private keys encrypted using AES-128. All pre-shared and symmetric keys are stored in encrypted form using AES-128 using the intel Xeon AES-NI encryption. The TOE is configured to be in FIPS mode by entering the 'fipsconfig' command at the CLI. During the FIPS mode setup, an authorized administrator is able to select the option to have all passwords and keys encrypted. In addition, there is a sub-option using the 'saveconfig' command and the save config dialog in the GUI to encrypt the passwords and keys. There are no administrative interfaces available that allow passwords to be viewed.
FPT_STM.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. The TOE can optionally be set to receive clock updates from an NTP server using the 'ntpconfig' command or via the GUI in the Time Zone or Time Settings page from the System Administration menu. 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. The time information is also used in setting the system time and administrative session timeout. The time can be configured using the CLI commands: settime and settz. In the GUI, the time can be configured under the Time Zone or Time Settings page from the System Administration menu.
FPT_TST_EXT.1	The TOE runs a suite of self-tests during initial start-up to verify its correct operation also referred to as Power on Self-Tests (POST). The POST verifies the integrity of the software and ensures all cryptographic software. If any of the tests fail, the Authorized Administrator will have to log into the CLI to determine which test failed and why. During the system bootup process (power on or reboot), all the Power on Startup Test (POST) components for all the cryptographic modules perform the POST for the corresponding component (hardware or software). These tests include: • Data integrity of the message queue store journal • checks a known virus (spam) stamp against anti-virus (-spam) engines

TOE SFRs	How the SFR is Met
	• AES Known Answer Test • AES-CCM Known Answer Test • AES-GCM Known Answer Test • AES-CMAC Known Answer Test • Triple-DES Known Answer Test • DSA Sign/Verify Test • RSA Signature Known Answer Test • ECDSA Sign/Verify Test • RNG Known Answer Test • HMAC Known Answer Test (performed for each supported SHA) • SHA-1 Known Answer Test • SHA-2 Known Answer Test (includes SHA-224, SHA-256, SHA-384 and SHA-512) • Software Integrity Test If any component reports failure for the POST, the system crashes and appropriate information is displayed on the screen, and an alert is sent to an administrative email each time a self test fails for any reason and a failed part of the functionality is disabled until a problem resolution has been accomplished. All ports are blocked from moving to forwarding state during the POST. If all components of all modules pass the POST, the system is placed in FIPS PASS state and ports are allowed to forward data traffic. These tests are sufficient to verify that the correct version of the TOE software is running as well as that the cryptographic operations are all performing as expected. Both of these functions are required to ensure that the TOE is operating as expected and data that the user expects to be encrypted in not transferred in plaintext.
FPT_TUD_EXT.1	An Authorized Administrator can query the currently executing software version as well as the last installed version via the CLI and GUI. An authorized administrator can either manually download the updates or ESA can automatically download the updates when "automated updates" has been configured. Updates can be downloaded directly from the Cisco Update Servers as well as from an offline update server. Both an administrator and the TOE can check to see if an update is available from Cisco. If automated updates have been configured by an Authorized Administrator, then the TOE downloads the update which is an encrypted file and a config file with a hash of the update. The TOE decrypts the update. Verification of the authenticity of the image and software updates is done in an automated manner. ESA automatically compares the hash received via the configuration file to the hash computed for the product update using SHA-384. If there is a checksum mismatch, the update will not be installed. Attempts to perform an illegitimate update onto the system will be logged into updater logs at INFO level. The sample log line will look as follows: `Wed Dec 11 05:50:07 2013 Info: repeng SHA384 Mismatch`

TOE SFRs	How the SFR is Met
FTA_SSL_EXT.1 FTA_SSL.3	An administrator can configure maximum inactivity times individually for both the CLI and GUI. An authorized administrator can specify how long a user can be logged into the Email Security appliance’s Web UI before AsyncOS logs the user out due to inactivity by default it is set to 30 minutes. This Web UI session timeout applies to all users, including administrators, and it is used for both HTTP and HTTPS sessions. For this purposes of the evaluated configuration only HTTPS is permitted. Once AsyncOS logs a user out, the appliance redirects the user’s web browser to the login page. Likewise an authorized administrator can specify how long a user can be logged into the Email Security appliance’s CLI before AsyncOS logs the user out due to inactivity. If a local user session is inactive for a configured period of time, the session will be locked and will require re-authentication to unlock the session. If a remote user session is inactive for a configured period of time, the session will be terminated and will require authentication to establish a new session.
FTA_SSL.4	An administrator is able to exit out of both the CLI and GUI administrative sessions. An authorized administrator can log out of the CLI with the 'exit' command. The Web UI also has a logout option via the drop-down menu.
FTA_TAB.1	The TOE displays an authorized administrator specified banner on the CLI and GUI management interfaces prior to allowing any administrative access to the TOE. This is applicable for both the CLI and GUI being accessed locally and remotely.
FTP_ITC.1	The TOE protects communications with the syslog server using SSHv2. SSHv2 uses a keyed hash as defined in FCS_SSHC_EXT.1.6. This protects the data from modification by hashing the data and verifying the hash on receipt of the data. This ensures that the data has not been modified in transit. In addition, encryption of the data as defined in FCS_SSHC_EXT.1.4 is provided to ensure the data is not disclosed in transit. SCP Push is used for sending audit logs securely over SSHv2 to a syslog server. This method periodically pushes log files to a remote Syslog server. It requires an SSH server on the Syslog Server using the SSHv2 protocol. The subscription requires a username, SSH key, and destination directory on the remote computer. Log files are transferred based on a rollover schedule set by an Authorized Administrator.
FTP_TRP.1	All remote administrative communications take place over a secure encrypted SSHv2 for the CLI or TLS/HTTPS for the GUI sessions. The SSHv2 session is encrypted using AES encryption. The remote users are able to initiate SSHv2 communications with the TOE for secure CLI access. TLS/HTTPS is used to secure the communications with the TOE and remote web browser for secure GUI access.

7 ANNEX A: KEY ZEROIZATION

7.1 Key Zeroization

The following table describes the key zeroization referenced by FCS_CKM.4 provided by the TOE.

Table 19: TOE Key Zeroization

Name	**Description**	**Zeroization**
Diffie-Hellman Shared Secret	The value is zeroized after it has been given back to the consuming operation. The value is overwritten by 0's.	Automatically after completion of DH exchange. Overwritten with: 0x00
Diffie Hellman private exponent	The function returns the value to the RP and then calls the function to perform the zeroization of the generated key pair (p_dh_kepair) and then calls the standard Linux free (without the poisoning). These values are automatically zeroized after generation and once the value has been provided back to the actual consumer.	Zeroized upon completion of DH exchange. Overwritten with: 0x00
SSH Private Key	Once the function has completed the operations requiring the RSA key object, the module over writes the entire object (no matter its contents) using memset. This overwrites the key with all 0's.	Zeroized upon deletion of the SSH public/private key pair when no longer needed. Overwritten with: 0x00
AES Key	The results zeroized using the poisoning in free to overwrite the values with 0x00. This is called by the ssh_close/tls_close function when a session is ended.	Automatically when the SSH/TLS session is terminated. Overwritten with: 0x00

ANNEX B: REFERENCES

The following documentation was used to prepare this ST:

Table 20: References

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, CCMB-2012-009-001
[CC_PART2]	Common Criteria for Information Technology Security Evaluation – Part 2: Security functional components, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-002
[CC_PART3]	Common Criteria for Information Technology Security Evaluation – Part 3: Security assurance components, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-003
[CEM]	Common Methodology for Information Technology Security Evaluation – Evaluation Methodology, dated September 2012, version 3.1, Revision 4, CCMB-2012-009-004
[NDcPP]	collaborative Protection Profile for Network Devices, version 1.0, 27-Feb-2015
[800-38A]	NIST Special Publication 800-38A Recommendation for Block 2001 Edition Recommendation for Block Cipher Modes of Operation Methods and Techniques December 2001
[800-56A]	NIST Special Publication 800-56A Rev 2, May 2013 Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography
[800-56B]	NIST Special Publication 800-56B Recommendation for Pair-Wise, August 2009 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 186-2]	FIPS PUB 186-2 Federal Information Processing Standards Publication 2000 January 27
[FIPS PUB 186-3]	FIPS PUB 186-3 Federal Information Processing Standards Publication Digital Signature Standard (DSS) June, 2009
[FIPS PUB 186-4]	FIPS PUB 186-4 Federal Information Processing Standards Publication Digital Signature Standard (DSS) July 2013
[FIPS PUB 198-1]	Federal Information Processing Standards Publication The Keyed-Hash Message Authentication Code (HMAC) July 2008
[NIST SP 800-90A Rev 1]	NIST Special Publication 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators January 2015
[FIPS PUB 180-3]	FIPS PUB 180-3 Federal Information Processing Standards Publication Secure Hash Standard (SHS) October 2008

ANNEX C: TECHNICAL DECISIONS

The following Technical Decisions apply to the NDcPPv1.0:

Table 21 NDcPP Technical Decisions

ID	Description	PP Relates to	Date
TD0201	NIT Technical Decision for Use of intermediate CA certificates and certificate hierarchy depth	CPP_ND_V1.0, CPP_FW_V1.0, ND SD v1.0, FIA_X509_EXT.1.1, FIA_X509_EXT.1.2	2017.05.03
TD0200	NIT Technical Decision for Password authentication for SSH clients	CPP_ND_V1.0, CPP_FW_V1.0, ND SD v1.0, FCS_SSHC_EXT.1.2	2017.05.01
TD0199	NIT Technical Decision for Elliptic Curves for Signatures	CPP_ND_V1.0, CPP_FW_V1.0, FCS_COP.1	2017.05.01
TD0195	NIT Technical Decision Making DH Group 14 optional in FCS_IPSEC_EXT.1.11	CPP_ND_V1.0, CPP_FW_V1.0, FCS_IPSEC_EXT.1.11	2017.04.21
TD0191	NIT Technical Decision for Using secp521r1 for TLS communication	CPP_ND_V1.0, CPP_FW_V1.0, FCS_TLSS_EXT.1.3. FCS_TLSS_EXT.2.3	2017.04.10
TD0189	NIT Technical Decision for SSH Server Encryption Algorithms	NDcPP V1.0, FWcPP V1.0, FCS_SSHC_EXT.1.4, FCS_SSHS_EXT.1.4	2017.04.10
TD0188	NIT Technical Decision for Optional use of X.509 certificates for digital signatures	NDcPP V1.0, FWcPP V1.0, FPT_TUD_EXT.1, FPT_TUD_EXT.2	2017.04.10
TD0187	NIT Technical Decision for Clarifying FIA_X509_EXT.1 test 1	NDcPP V1.0, FWcPP V1.0, ND SD v1.0, FIA_X509_EXT.1.1	2017.04.10
TD0186	NIT Technical Decision for Applicability of X.509 certificate testing to IPsec	NDcPP V1.0, FWcPP V1.0, FIA_X509_EXT.1.1	2017.04.10
TD0185	NIT Technical Decision for Channel for Secure Update.	NDcPP V1.0, FWcPP V1.0, FPT_TUD_EXT.1, FTP_ITC.1	2017.04.10
TD0184	NIT Technical Decision for Mandatory use of X.509 certificates	NDcPP V1.0, FWcPP V1.0, FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3	2017.04.10
TD0182	NIT Technical Decision for Handling of X.509 certificates related to ssh-rsa and remote comms.	NDcPP V1.0, FWcPP V1.0, FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3	2017.04.10
TD0181	NIT Technical Decision for Self-testing of integrity of firmware and software.	NDcPP V1.0, FWcPP V1.0, ND SD V1.0, FPT_TST_EXT.1	2017.04.10
TD0170	NIT Technical Decision for SNMPv3 Support	CPP_ND_V1.0, CPP_FW_V1.0, FTP_TRP.1	2017.04.04
TD0169	NIT Technical Decision for Compliance to RFC5759 and RFC5280 for using CRLs	CPP_ND_V1.0, CPP_FW_V1.0, FIA_X509_EXT.1.1	2017.04.04
TD0168	NIT Technical Decision for Mandatory requirement for CSR generation	CPP_ND_V1.0, CPP_FW_V1.0, FIA_X509_EXT.3	2017.04.04
TD0167	NIT Technical Decision for Testing SSH 2^28 packets	CPP_ND_V1.0, CPP_FW_V1.0, ND SD V1.0, FCS_SSHC_EXT.1.8, FCS_SSHS_EXT.1.8, FMT_SMF.1	2017.04.04
TD0164	NIT Technical Decision for Negative testing for additional ciphers for SSH	CPP_ND_V1.0, ND SD V1.0, FCS_SSHC_EXT.1.4, FCS_SSHS_EXT.1.4	2017.03.21

TD0160	NIT Technical Decision for Transport mode and tunnel mode in IPSEC communications	CPP_ND_V1.0, CPP_FW_V1.0, FCS_IPSEC_EXT.1.3	2017.03.08
TD0156	NIT Technical Decision for SSL/TLS Version Testing in the NDcPP v1.0 and FW cPP v1.0	CPP_ND_V1.0, CPP_FW_V1.0, ND SD V1.0, FCS_TLSS_EXT1.2, FCS_TLSS_EXT.2.2	2017.03.15
TD0155	NIT Technical Decision for TLSS tests using ECDHE in the NDcPP v1.0.	CPP_ND_V1.0, ND SD V1.0, FCS_TLSS_EXT.1.3, FCS_TLSS_EXT.2.3	2017.03.01
TD0154	NIT Technical Decision for Versions of TOE Software in the NDcPP v1.0 and FW cPP v1.0	CPP_ND_V1.0, CPP_FW_V1.0, FPT_TUD_EXT.1	2017.03.01
TD0153	NIT Technical Decision for Auditing of NTP Time Changes in the NDcPP v1.0 and FW cPP v1.0	CPP_ND_V1.0, CPP_FW_V1.0, FAU_GEN.1, FPT_STM.1	2017.03.01
TD0152	NIT Technical Decision for Reference identifiers for TLS in the NDcPP v1.0 and FW cPP v1.0	CPP_ND_V1.0, CPP_FW_V1.0, ND SD V1.0, FCS_TLSC_EXT.1, FCS_TLSC_EXT.2	2017.03.01
TD0151	NIT Technical Decision for FCS_TLSS_EXT Testing - Issue 1 in NDcPP v1.0.	CPP_ND_V1.0, ND SD V1.0, FCS_TLSS_EXT.1, FCS_TLSS_EXT.2	2017.03.01
TD0150	NIT Technical Decision for Removal of SSH re-key audit events in the NDcPP v1.0 and FW cPP v1.0	CPP_ND_V1.0, CPP_FW_V1.0, FAU_GEN.1	2017.03.01
TD0143	NIT Technical Decision for Failure testing for TLS session establishment in NDcPP and FWcPP	FCS_TLSS_EXT.1.1, CPP_ND_V1.0, CPP_FW_V1.0,	2017.01.18
TD0130	NIT Technical Decision for Requirements for Destruction of Cryptographic Keys	CPP_ND_V1.0, CPP_FW_V1.0, FCS_CKM.4	2016.12.08
TD0126	NIT Technical Decision for TLS Mutual Authentication	CPP_ND_V1.0, CPP_FW_V1.0, FTP_ITC.1, FCS_TLSC_EXT.1	2016.11.30
TD0125	NIT Technical Decision for Checking validity of peer certificates for HTTPS servers	CPP_ND_V1.0, CPP_FW_V1.0, , ND SD v1.0, FCS_HTTPS_EXT.1.2	2016.11.15
TD0117	NIT Technical Decision for FIA_X509_EXT.1.1 Requirement in NDcPP	CPP_ND_V1.0, CPP_ND_V1.0, FIA_X509_EXT, FPT_TST_EXT, FPT_TUD_EXT	2016.10.13
TD0116	NIT Technical Decision for a Typo in reference to RSASSA-PKCS1v1_5 in NDcPP and FWcPP	CPP_ND_V1.0, CPP_FW_V1.0, FCS_COP.1.1(2)	2016.10.13