Symantec Corporation
Security Analytics S500 Appliances

Models: SA-S500-10-CM, SA-S500-20-FA, SA-S500-30-FA, SA-S500-40-FA
Firmware Version: 7.2.4

Security Target

Document Version: 0.10

Contact Information

Americas:
Symantec Corporation
350 Ellis Street
Mountain View, CA 94043
www.symantec.com

Table of Contents

List of Figures

List of Tables

1. Introduction

1.1 Purpose

This is a Non-Proprietary Cryptographic Module Security Target for the Security Analytics S500 Appliance (SA-S500-10-CM, SA-S500-20-F, SA-S500-30-F, SA-S500-40-F; 7.2.4) from Symantec Corporation. This Non-Proprietary Security Target describes how the Security Analytics S500 Appliance meets the security requirements for the Network Device Collaborative Protection Profile. More information can be found at https://www.niap-ccevs.org/Profile/Info.cfm?id=372.

1.2 Security Target and TOE References

Table 1 below shows the ST and TOE references.

Table 1 ST and TOE References

ST Title	Symantec Corporation Security Analytics S500 Appliances Security Target
ST Version	0.10
ST Author	Acumen Security, LLC.
ST Publication Date	December 10, 2018
TOE Reference	Symantec Corporation Security Analytics S500 Appliances
TOE Software Version	Version 7.2.4
TOE Build Number	Build 45794
TOE Hardware Version	SA-S500-10-CM, SA-S500-20-FA, SA-S500-30-FA, SA-S500-40-FA
TOE developer	Symantec Corporation

1.3 Product Overview

The Security Analytics Appliances (SA-S500-10-CM, SA-S500-20-FA, SA-S500-30-FA, SA-S500-40-FA) are part of Symantec's Security Platform's Incident Response and Forensics solutions. The turnkey, pre-configured appliances harness the Security Analytics software to capture, index and classify all network traffic (including full packets) in real time. This data is stored in an optimized file system for rapid analysis, instant retrieval and complete reconstruction to support all your incident response activities. The appliances can be deployed anywhere in the network: at the perimeter, in the core, in a 10 GbE backbone, or at a remote link to deliver clear, actionable intelligence for swift incident response and resolution and real-time network forensics.

Security Analytics helps you visualize and analyze network data and uncover specific network activity – without requiring specific knowledge of networking protocols and packet analysis methods. Its powerful features let you locate and reconstruct specific communication flows, as well as network and user activities, within seconds. The platform does this by classifying captured network traffic packets and identifying meaningful data flows. A flow is the collection of packets that comprises a single communication between two specific network entities. Within a particular data flow, you can then identify and examine network artifacts such as image files, Word documents, emails, and video, as well as executable files, HTML files,

and more. Security Analytics also allows you to reconstruct HTML pages, emails, and instant messaging conversations.

Security Analytics also provides the ability to do real-time, policy-based artifact extraction, and is not limited to any specific operating system (OS) environment. Extracted artifacts can be automatically placed in centralized network repositories for analysis by superior forensics tools within Security Analytics. These artifacts are hashed and stored for future retrospection on newly discovered malware variants and provide a method to understand relatedness to preexisting hashes. The Central Manager Appliance (SA-S500-10-CM) facilitates federated queries on hundreds of Security Analytics Forensic Devices (SA-S500-20-F, SA-S500-30-F, SA-S500-40-F) to provide a 360-degree view of activity across the entire enterprise network including perimeter, data centers, and remote offices.

1.4 TOE Overview

The TOE is a network security analytic appliance that can be deployed anywhere in a network to provide a clear view of the installed network. The TOE supports mutual authentication with an audit server as a TLS client. In addition, the TOE can rest in different areas of the network, such as on the perimeter, in the core, in a backbone or at a remote link to deliver clear, actionable intelligence. The TOE also provides real-time, policy-based artifact extraction, and is not limited to any specific operating system (OS).

The following table identifies the physical characteristics of the TOE:

Table 2 Physical Characteristics

		SA-S500-10-CM	SA-S500-20-FA	SA-S500-30-FA	SA-S500-40-FA
Ports	**Front**	• 1 USB Port			
	Rear	• 4 Ethernet Ports • 1 MGMT Port • 1 Serial Port • 1 BMC MGMT Port • 1 USB Port	• 2 Ethernet 10Gig ports • 1 MGMT Port • 1 BMC MGMT Port • 4x12Gbps SAS3 Port • 2x SX/SR Fiber Channel Port • 1 Serial Port • 1 USB Port	• 2 Ethernet 10Gig ports • 4 Ethernet Ports • 1 MGMT Port • 1 BMC MGMT Port • 2x12Gbps SAS3 Port • 4x SX/SR Fiber Channel Port • 1 Serial Port • 1 USB Port	• 2 Ethernet 10Gig ports • 4 Ethernet Ports • 1 MGMT Port • 1 BMC MGMT Port • 2x12Gbps SAS3 Port • 4x SX/SR Fiber Channel Port • 1 Serial Port • 1 USB Port
Enclosure	**Front**	• 2 RU • 2 LEDS • 1 LCD • 6 control buttons			
	Rear	• 1 Power Switch • 6 Ethernet Speed LEDs • 6 Ethernet Activity LEDs • 2 AC Power LEDs	• 1 Power Switch • 4 Ethernet Speed LEDs • 4 Ethernet Activity LEDs • AC Power LEDs	• 1 Power Switch • 6 Ethernet Speed LEDs • 6 Ethernet Activity LEDs • 2 AC Power LEDs	• 1 Power Switch • 6 Ethernet Speed LEDs • 6 Ethernet Activity LEDs • 2 AC Power LEDs
Power Supply		• 2			•
Software		Security Analytics Software Version 7.2.4			

1.5 TOE Evaluated Configuration

The TOE evaluated configuration is comprised of at least one of the following: SA-S500-10-CM, SA-S500-20-FA, SA-S500-30-FA, or SA-S500-40-FA. The evaluated configuration also supports the following external IT entities;

Table 3 IT Environment Components

Component	Required	Usage/Purpose Description for TOE performance
Remote Management Workstation (GUI).	Yes	This includes any IT Environment Management workstation with a web browser installed that is used by the TOE administrator to support TOE administration through HTTPS and TLS protected channels.
Remote Management Workstation (CLI).	Yes	This includes any IT Environment Management workstation with an SSH client installed that is used by the TOE administrator to support TOE administration through SSH protected channels.
Local Management Workstation (CLI).	Yes	This includes any IT Environment Management workstation with a local CLI support that is used by the TOE administrator to support TOE administration through a direct connection.
Certificate Authority	Yes	The CA is used in support of certificate validation operations.
Syslog Server	Yes	The syslog audit server is used for remote storage of audit records that have been generated by and transmitted from the TOE.
CRL Server	Yes	The CRL server is used to in support of certificate revocation testing.

1.6 TOE Architecture

1.6.1 Physical Boundaries

The TOE is a hardware and software solution that is comprised of the network device and its 3 configurations described in section 1.4. The diagram below depicts the evaluated configuration. The red rectangle represents the physical boundary of the TOE.

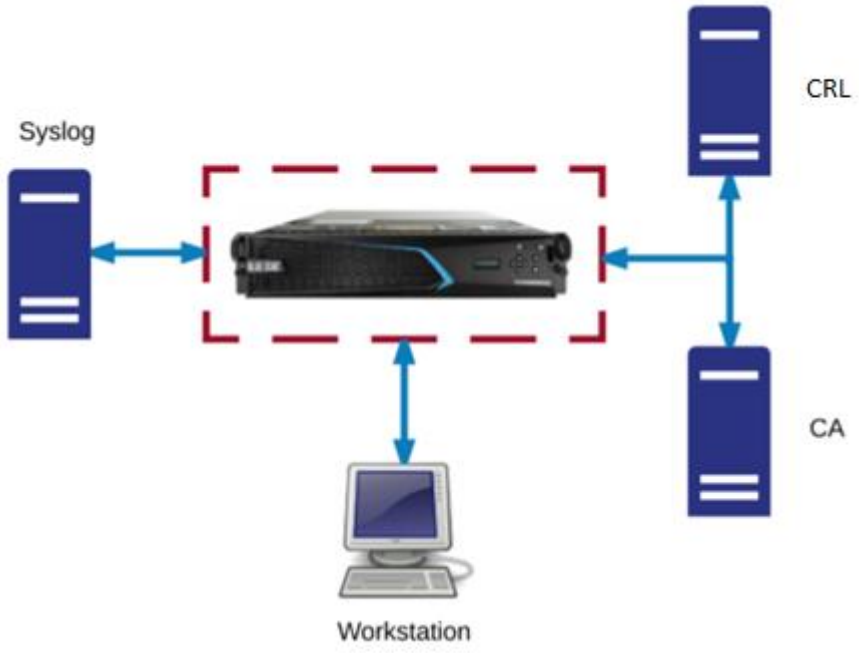

Figure 1 Physical Boundary

The TOE hardware includes the SA-S500-10-CM, SA-S500-20-FA, SA-S500-30-FA, and SA-S500-40-FA. Section 1.4 provides details on the number of ports and LED indicators on each of these devices. These devices come pre-installed with the TOE Software version 7.2.4 as identified in Section 1.2 above. The IPv4 network on which the TOE resides is considered part of the environment.

In addition, as part of the evaluation, the TOE IT environment includes the use of a Certificate Authority (CA), Syslog Server, and Certificate Revocation List (CRL) service. This is shown in Figure 1 above.

For proper configuration of the TOE into the evaluated configuration, the following guidance documents are available:

- Symantec Corporate Security Analytics S500 Appliances Common Criteria Administrative Guidance Document
- Security Analytics 7.2.3 Administration and Central Manager Guide
- Security Analytics 7.2.3 Reference Guide

1.6.2 Logical Boundaries

The TOE provides several types of security functionalities, including.

- Security Audit
- Cryptography Support

- Identification & Authentication
- Security Management
- Protection of the TSF
- TOE Access
- Trusted Path/Channel

These features are described in more detail in the subsections below. In addition, the TOE implements all RFCs of the Collaborative Protection Profile for Network Devices necessary to satisfy testing/ assurance measures prescribed therein.

1.6.2.1 Security Audit

The Network Appliances provide extensive auditing capabilities. The TOE generates a comprehensive set of audit logs that identify specific TOE operations. For each event, the TOE records the date and time of each event, the type of event, the subject identity, and the outcome of the event. Auditable events include:

- Start-up of the TOE from both cold boot and reboot,
- Shutdown of the TOE (when shut down from the local CLI, Remote CLI, and GUI),
- All administrative actions (both security relevant and non-security relevant) from the local CLI, Remote CLI, and GUI,
- Remote administrative HTTPS/TLS connection establishment,
- Remote administrative HTTPS/TLS connection closure,
- Errors during Remote administrative HTTPS/TLS connection establishment,
- Remote administrative SSH connection establishment,
- Remote administrative SSH connection closure,
- Errors during Remote administrative SSH connection establishment,
- Generation of self-signed certificates,
- Import of certificates,
- Deletion of certificates,
- Successful authentication attempts (from the local CLI, Remote CLI, and GUI),
- Unsuccessful authentication attempts (from the local CLI, Remote CLI, and GUI),
- Unsuccessful certificate validation for the presence of the basicConstraints extension missing,
- Unsuccessful certificate validation for the CA flag is set to TRUE for all CA certificates,
- Unsuccessful certificate validation for trust chain verification failure,
- Unsuccessful certificate validation for revocation status,
- All attempts to update the TOE software,
- Changes to time,
- Start of a local administrative session,
- End of a local administrative session,
- Administration session timeout (from the local CLI, Remote CLI, and GUI).

The TOE is configured to transmit its audit messages to an external syslog server. Communication with the syslog server is protected using TLS.

The logs for all the appliances can be viewed via the remote GUI interface or through the CLI. The records include the date/time the event occurred, the event/type of event, the user ID associated with the event, and additional information of the event and its success and/or failure.

1.6.2.2 Cryptographic Support

The TOE provides cryptographic support for the following features,

- TLSv1.1, TLSv1.2 and HTTPS connectivity with the following entities:

  - Management Web Browser,
  - Audit Server.
- SSH connectivity with the following entities:
  - Management SSH Client.
- Secure software update

The Cryptographic services provided by the TOE are described below;

Table 4 Provided Cryptography

Cryptographic Method	Use within the TOE
AES	• TLS Traffic Encryption/Decryption • SSH Traffic Encryption/Decryption
RSA	• TLS Session Establishment • SSH Session Establishment • Software Upgrade
SP800-90A	• TLS Session Establishment • SSH Session Establishment
SHS	• Used to provide TLS traffic integrity verification • Used to provide SSH traffic integrity verification
HMAC-SHS	• Used to provide TLS traffic integrity verification • Used to provide SSH traffic integrity verification
SP800-56A	• TLS Session Establishment • SSH Session Establishment
SP800-135rev1	• TLS Session Key Derivation • SSH Session Key Derivation

1.6.2.3 Identification and Authentication

The TOE provides authentication services for administrative users to connect to the TOEs administrator interfaces (local CLI, remote CLI, and remote GUI). The TOE requires Authorized Administrators to authenticate prior to being granted access to any of the management functionality. In the Common Criteria evaluated configuration, the TOE is configured to require a minimum password length of 15 characters. The TOE provides administrator authentication against a local user database. Password-based authentication can be performed on any TOE administrative.

1.6.2.4 Security Management

The TOE provides secure administrative services for management of general TOE configuration and the security functionality provided by the TOE. Management can take place over a variety of interfaces including:

- Local console command line administration;
- Remote CLI administration via SSH;
- Remote GUI administration via HTTPS/TLS.

.

All administration functions can be accessed via, remote CLI, remote GUI or via a direct connection to the TOE. The TOE provides the ability to securely manage the below listed functions;

- 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.

1.6.2.5 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 Administrators. The TOE prevents reading of cryptographic keys and passwords. Additionally, the TOE software (TBD) is custom-built for the appliance. 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. Finally, the TOE performs testing to verify correct operation of the security appliances themselves. The TOE verifies all software updates via digital signature (4096-bits/SHA-512) and requires administrative intervention prior to the software updates being installed on the TOE to avoid the installation of unauthorized software.

1.6.2.6 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 displays an Authorized Administrator specified banner on both the CLI and GUI management interfaces prior to allowing any administrative access to the TOE.

1.6.2.7 Trusted Path/Channels

The TOE supports several types of secure communications, including,

- Trusted paths with remote administrators over SSH,
- Trusted paths with remote administrators over TLS/HTTPS,
- Trusted channels with remote IT environment audit servers over TLS.

2. Conformance Claims

2.1 CC Conformance

This TOE is conformant to:

- Common Criteria for Information Technology Security Evaluations Part 1, Version 3.1, Revision 5, April 2017
- Common Criteria for Information Technology Security Evaluations Part 2, Version 3.1, Revision 5, April 2017: Part 2 extended
- Common Criteria for Information Technology Security Evaluations Part 3, Version 3.1, Revision 5, April 2017: Part 3 conformant

2.2 Protection Profile Conformance

The TOE is conformant to:

- Collaborative Protection Profile for Network Devices, Version 1.0, 27 February 2015 [NDcPP].

In addition to the Protection Profile identified above, the following NIT technical decisions have been applied from the Network Device Collaborative Protection Profile:

- TD0291/NIT Technical Decision for DH14 and FCS_CKM.1
- TD0290/NIT Technical Decision for physical interruption of trusted path/channel.
- TD0289/NIT Technical Decision for FCS_TLSC_EXT.x.1 Test 5e
- TD0281/NIT Technical Decision for Testing both thresholds for SSH rekey
- TD0257/NIT Technical Decision for Updating FCS_DTLSC_EXT.x.2/FCS_TLSC_EXT.x.2 Tests 1-4
- TD0256/NIT Technical Decision for Handling of TLS connections with and without mutual authentication
- TD0255/NIT Technical Decision for TLS Server Tests - Issue 3: Verification of application of encryption
- TD0235/NIT Technical Decision adding DH group 14 to the selection in FCS_CKM.2
- TD0228/NIT Technical Decision for CA certificates - basicConstraints validation
- TD0227/NIT Technical Decision for TOE acting as a TLS Client and RSA key generation
- TD0226/NIT Technical Decision for TLS Encryption Algorithms
- TD0201/NIT Technical Decision for Use of intermediate CA certificates and certificate hierarchy depth
- TD0199/NIT Technical Decision for Elliptic Curves for Signatures
- TD0189/NIT Technical Decision for SSH Server Encryption Algorithms
- TD0188/NIT Technical Decision for Optional use of X.509 certificates for digital signatures
- TD0187/NIT Technical Decision for Clarifying FIA_X509_EXT.1 test 1
- TD0186/NIT Technical Decision for Applicability of X.509 certificate testing to IPsec
- TD0185/NIT Technical Decision for Channel for Secure Update.
- TD0184/NIT Technical Decision for Mandatory use of X.509 certificates
- TD0183/NIT Technical Decision for Use of the Supporting Document
- TD0182/NIT Technical Decision for Handling of X.509 certificates related to ssh-rsa and remote comms.
- TD0181/NIT Technical Decision for Self-testing of integrity of firmware and software.

- TD0170/NIT Technical Decision for SNMPv3 Support
- TD0169/NIT Technical Decision for Compliance to RFC5759 and RFC5280 for using CRLs
- TD0168/NIT Technical Decision for Mandatory requirement for CSR generation
- TD0167/NIT Technical Decision for Testing SSH 2^28 packets
- TD0165/NIT Technical Decision for Sending the ServerKeyExchange message when using RSA
- TD0164/NIT Technical Decision for Negative testing for additional ciphers for SSH
- TD0156/NIT Technical Decision for SSL/TLS Version Testing in the NDcPP v1.0 and FW cPP v1.0
- TD0155/NIT Technical Decision for TLSS tests using ECDHE in the NDcPP v1.0.
- TD0154/NIT Technical Decision for Versions of TOE Software in the NDcPP v1.0 and FW cPP v1.0
- TD0153NIT Technical Decision for Auditing of NTP Time Changes in the NDcPP v1.0 and FW cPP v1.0
- TD0152/NIT Technical Decision for Reference identifiers for TLS in the NDcPP v1.0 and FW cPP v1.0
- TD0151/NIT Technical Decision for FCS_TLSS_EXT Testing - Issue 1 in NDcPP v1.0.
- TD0150/NIT Technical Decision for Removal of SSH re-key audit events in the NDcPP v1.0 and FW cPP v1.0
- TD0143/NIT Technical Decision for Failure testing for TLS session establishment in NDcPP and FWcPP
- TD0130/NIT Technical Decision for Requirements for Destruction of Cryptographic Keys
- TD0126/NIT Technical Decision for TLS Mutual Authentication
- TD0125/NIT Technical Decision for Checking validity of peer certificates for HTTPS servers
- TD0117/NIT Technical Decision for FIA_X509_EXT.1.1 Requirement in NDcPP
- TD0116/NIT Technical Decision for a Typo in reference to RSASSA-PKCS1v1_5 in NDcPP and FWcPP
- TD0114/NIT Technical Decision for Re-Use of FIPS test results in NDcPP and FWcPP
- TD0113/NIT Technical Decision for testing and trusted updates in the NDcPP v1.0 and FW cPP v1.0
- TD0112/NIT Technical Decision for TLS testing in the NDcPP v1.0 and FW cPP v1.0.
- TD0111/NIT Technical Decision for third party libraries and FCS_CKM.1 in NDcPP and FWcPP
- TD0096/NIT Technical Interpretation regarding Virtualization
- TD0095/NIT Technical Interpretations regarding audit, random bit generation, and entropy in NDcPP
- TD0094/NIT Technical Decision for validating a published hash in NDcPP
- TD0090/NIT Technical Decision for FMT_SMF.1.1 Requirement in NDcPP

The following NIT technical decisions applicable to the Network Device Collaborative Protection Profile are not applicable to this ST for the reasons stated:

- TD0262/NIT Technical Decision for TLS server testing - Empty Certificate Authorities list (archived)
- TD0225/NIT Technical Decision for Make CBC cipher suites optional in IPsec (applicable to FCS_IPSEC_EXT.1, which is not included in this ST)
- TD0224/NIT Technical Decision Making DH Group 14 optional in FCS_IPSEC_EXT.1.11 (applicable to FCS_IPSEC_EXT.1, which is not included in this ST)
- TD0223/NIT Technical Decision for "Expected" vs "unexpected" DNs for IPsec Communications (applicable to FCS_IPSEC_EXT.1, which is not included in this ST)
- TD0200/NIT Technical Decision for Password authentication for SSH clients (applicable to FCS_SSHC_EXT.1, which is not included in this ST)

- TD0160/NIT Technical Decision for Transport mode and tunnel mode in IPSEC communications (applicable to FCS_IPSEC_EXT.1, which is not included in this ST)
- TD0115/NIT Technical Decision for Transport mode and tunnel mode in IPsec communication in NDcPP and FWcPP (applicable to FCS_IPSEC_EXT.1, which is not included in this ST)
- TD0093/NIT Technical Decision for FIA_X509_EXT.1.1 Requirement in NDcPP (superceded by TD0117)

In all instance in which a Technical Decision was found to be applicable based on the selection of SFRs within the Security Target, the appropriate revisions were made.

2.3 Conformance Rationale

This Security Target claims exact conformance to the following protection profile:

- Collaborative Protection Profile for Network Devices, Version 1.0, 27 February 2015 [NDcPP].

The Security Target for the TOE contains the security problem definition, security objectives and security requirements taken direct from the Protection Profile and performing only the operations defined there.

3. Security Problem Definition

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

3.1 Threats

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

3.1.1 Communications with the Network Device

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

Authorized communication includes network traffic allowable by policy destined to and originating from the network device as it was designed and intended. This includes critical network traffic, such as network device administration and communication with an authentication or audit logging server, which requires a secure channel to protect the communication. The security functionality of the network device includes the capability to ensure that only authorized communications are allowed and the capability to provide a secure channel for critical network traffic. Any other communication is considered unauthorized communication.

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

3.1.1.1 T.UNATHORIZED_ADMINISTRATOR_ACCESS

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

3.1.1.2 T.WEAK_CRYPTOGRAPHY

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

3.1.1.3 T.UNTRUSTED_COMMUNICATION_CHANNELS

Threat agents may attempt to target network devices that do not use standardized secure 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-the-middle attacks, replay attacks, etc. Successful attacks will result in loss of confidentiality and integrity of the critical network traffic, and potentially could lead to a compromise of the network device itself.

3.1.1.4 T.WEAK_AUTHENTICATION_ENDPOINTS

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

3.1.2 Valid Updates

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

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

3.1.2.1 T.UPDATE_COMPROMISE

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

3.1.3 Audited Activity

Auditing of network device activities is a valuable tool for administrators to monitor the status of the device. It provides the means for administrator accountability, security functionality activity reporting, reconstruction of events, and problem analysis. Processing performed in response to device activities may give indications of a failure or compromise of the security functionality. When indications of activity that impact the security functionality are not generated and monitored, it is possible for such activities to occur without administrator awareness. Further, if records are not generated and retained, reconstruction of the network and the ability to understand the extent of any compromise could be negatively affected. Additional concerns are the protection of the audit data that is recorded from alteration or unauthorized deletion. This could occur within the TOE, or while the audit data is in transit to an external storage device.

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

3.1.3.1 T.UNDETECTED_ACTIVITY

Threat agents may attempt to access, change, and/or modify the security functionality of the network device without administrator awareness. This could result in the attacker finding an avenue (e.g., misconfiguration, flaw in the product) to compromise the device and the administrator would have no knowledge that the device has been compromised.

3.1.4 Administrator and Device Credentials and Data

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

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

3.1.4.1 T.SECURITY_FUNCTIONALITY_COMPROMISE

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

3.1.4.2 T.PASSWORD_CRACKING

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

3.1.5 Device Failure

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

3.1.5.1 T.SECURITY_FUNCTIONALITY_FAILURE

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.

3.2 Assumptions

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

3.2.1 A.PHYSICAL_PROTECTION

The network device is assumed to be physically protected in its operational environment and not subject to physical attacks that compromise the security and/or interfere with the device's physical interconnections and correct operation. This protection is assumed to be sufficient to protect the device and the data it contains. As a result, the 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. [OE.PHYSICAL]

3.2.2 A.LIMITED_FUNCTIONALITY

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

3.2.3 A.NO_THRU_TRAFFIC_PROTECTION

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

3.2.4 A.TRUSTED_ADMINISTRATOR

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

3.2.5 A.REGULAR_UPDATES

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

3.2.6 A.ADMIN_CREDENTIALS_SECURE

The administrator's credentials (private key) used to access the network device are protected by the platform on which they reside. [OE.ADMIN_CREDENTIALS_SECURE]

3.3 Organizational Security Policy

An organizational security policy is a set of rules, practices, and procedures imposed by an organization to address its security needs. For the purposes of this cPP a single policy is described in the section below.

3.3.1 P.ACCESS_BANNER

The TOE shall display an initial banner describing restrictions of use, legal agreements, or any other appropriate information to which users consent by accessing the TOE. [FTA_TAB.1]

4. Security Objectives

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

4.1 Security Objectives for the Operational Environment

The following subsections describe objectives for the Operational Environment.

4.1.1 OE.PHYSICAL

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

4.1.2 OE.NO_GENERAL_PURPOSE

There are no general-purpose computing capabilities (e.g., compilers or user applications) available on the TOE, other than those services necessary for the operation, administration and support of the TOE.

4.1.3 OE.NO_THRU_TRAFFIC_PROTECTION

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

4.1.4 OE.TRUSTED_ADMIN

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

4.1.5 OE.UPDATES

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

4.1.6 OE.ADMIN_CREDENTIALS_SECURE

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

5. Security Requirements

This section identifies the Security Functional Requirements for the TOE and/or Platform. The Security Functional Requirements included in this section are derived from 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 made by PP author: Indicated with **bold** text and ~~strikethroughs~~, if necessary;
- Selection: Indicated with underlined text;
- Assignment within a Selection: Indicated with *italicized and underlined text*;
- Iteration: Indicated by appending the iteration number in parenthesis, e.g., (1), (2), (3).
- Where operations were completed in the PP itself, the formatting used in the PP has been retained.

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

5.2 TOE Security Functional Requirements

This section identifies the Security Functional Requirements for the TOE. The TOE Security Functional Requirements that appear below in Table 1 are described in more detail in the following subsections.

Table 5 TOE Security Functional Requirements and Auditable Events

Requirement	Auditable Events	Additional Audit Record Contents
FAU_GEN.1	None.	None.
FAU_GEN.2	None.	None.
FAU_STG_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 a HTTPS Session	Reason for failure
FCS_SSHS_EXT.1	Failure to establish an SSH session	Reason for failure
		Non-TOE endpoint of connection (IP Address)
FCS_TLSC_EXT.2	Failure to establish a TLS Session	Reason for failure
FCS_TLSS_EXT.1	Failure to establish a TLS Session	Reason for failure
FCS_RBG_EXT.1	None.	None.
FIA_PMG_EXT.1	None.	None.

Requirement	Auditable Events	Additional Audit Record Contents
FIA_UIA_EXT.1	All use of the identification and authentication mechanism.	Provided user identity, origin of the attempt authentication mechanism. (e.g., IP address)
FIA_UAU_EXT.2	All use of the identification and authentication mechanism.	Origin of the attempt (e.g., IP address).
FIA_UAU.7	None.	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_MOF.1(1)/Trusted Update	Any attempt to initiate a manual update	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 TUD EXT.1	Initiation of update; result of the update attempt (success or failure)	No additional information.
FPT TST EXT.1	None.	None.
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.
FTP_ITC.1	Initiation of the trusted channel. Termination of the trusted channel. Failure of the trusted channel functions.	Identification of the initiator and target of failed trusted channels establishment attempt.
FTP_TRP.1	Initiation of the trusted channel. Termination of the trusted channel. Failures of the trusted path functions.	Identification of the claimed user identity.

5.2.1 Class: Security Audit (FAU)

FAU_GEN.1 Audit Data Generation

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

a) Start-up and shut-down of the audit functions;
b) All auditable events for the not specified level of audit; and
c) *All administrative actions comprising:*
   - *Administrative login and logout (name of user account shall be logged if individual user accounts are required for administrators).*

- *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)*
- *[no other actions];*

d) [Specifically defined auditable events listed in Table 5].

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* 5.

FAU_GEN.2 User Identity Association

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

FAU_STG_EXT.1 Protected Audit Event Storage

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

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

FAU_STG_EXT.1.3 The TSF shall *[overwrite previous audit records according to the following rule: [the SA audit log rolls over the oldest audit messages using a series of 6 files, the oldest of the files being overwritten]*] when the local storage space for audit data is full.

5.2.2 Class: Cryptographic Support (FCS)

FCS_CKM.1 Cryptographic Key Generation

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

- ***RSA schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.3;***
- ***ECC schemes using "NIST curves" [P-256, P-384, P-521] that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.4;***
- ***FFC schemes using cryptographic key sizes of 2048-bit or greater that meet the following: FIPS PUB 186-4, "Digital Signature Standard (DSS)", Appendix B.1]***

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

FCS_CKM.2 Cryptographic Key Establishment

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

- ***RSA-based key establishment schemes that meets the following: NIST Special Publication 800-56B Revision 1, "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 Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography";***
- ***Finite field-based key establishment schemes that meets the following: NIST Special Publication 800-56A Revision 2, "Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography"***

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

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 [selection:*
  - *logically addresses the storage location of the key and performs a [single overwrite consisting of [zeroes]]]*

that meets the following: *No Standard.*

FCS_COP.1(1) Cryptographic Operation (AES Data Encryption/Decryption)
FCS_COP.1.1(1) The TSF shall perform *encryption/decryption* in accordance with a specified cryptographic algorithm *AES used in* [CBC, GCM*] mode* and cryptographic key sizes [*128 bits, 256 bits*] that meet the following: *AES as specified in ISO 18033-3, [CBC as specified in ISO 10116, GCM as specified in ISO 19772]*.

FCS_COP.1(2) Cryptographic Operation (Signature Generation and Verification)
FCS_COP.1.1(2) The TSF shall perform *cryptographic signature services (generation and verification)* in accordance with a specified cryptographic algorithm [

- *RSA Digital Signature Algorithm and cryptographic key sizes* ***(modulus)*** *[2048 bits, 3072 bits, and 4096 bits],*

]
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 RSASSA-PKCS1v1_5; ISO/IEC 9796-2, Digital signature scheme 2 or Digital Signature scheme 3,*

].

FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)
FCS_COP.1.1(3) 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*.

FCS_COP.1(4) Cryptographic Operation (Keyed Hash Algorithm)
FCS_COP.1.1(4) The TSF shall perform *keyed-hash message authentication* in accordance with a

specified cryptographic algorithm [*HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, HMAC-SHA-512*] and cryptographic key sizes [*160, 256, 384, 512-bits*] **and message digest sizes *[160, 256, 384, 512] bits*** that meet the following: *ISO/IEC 9797-2:2011, Section 7 "MAC Algorithm 2"*.

FCS_HTTPS_EXT.1 HTTPS Protocol

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

FCS_HTTPS_EXT.1.2 The TSF shall implement HTTPS using TLS.

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

FCS_RBG_EXT.1 Random Bit Generation
FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using [*HMAC_DRBG (any)*].

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

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 [*5647, 5656, 6668*].

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 [*1522*] bytes in an SSH transport connection are dropped.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms: [*aes128-cbc, aes256-cbc, AEAD_AES_128_GCM, AEAD_AES_256_GCM*].

FCS_SSHS_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_SSHS_EXT.1.6 The TSF shall ensure that the SSH transport implementation uses [*hmac-sha2-256, hmac-sha2-512*] and [*no other MAC algorithms*] as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that [*ecdh-sha2-nistp256*] and [*ecdh-sha2-nistp384, ecdh-sha2-nistp521*] 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.

FCS_TLSC_EXT.2 TLS Client Protocol with authentication
FCS_TLSC_EXT.2.1 The TSF shall implement [*TLS 1.2 (RFC 5246), TLS 1.1*

(RFC 4346)] supporting the following ciphersuites:

- *[*
  - *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
  - *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_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
  - *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
  - *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*
  - *Other - ECDHE-RSA-AES128-GCM-SHA256*
  - *Other - ECDHE-RSA-AES256-GCM-SHA384*

  ]

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

FCS_TLSC_EXT.2.3 The TSF shall only establish a trusted channel if the peer certificate is valid.

FCS_TLSC_EXT.2.4 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: [*secp256r1, secp384r1, secp521r1*] and no other curves.

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

FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1.1 The TSF shall implement [TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

- *[*
  - *TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268*
  - *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_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492*
  - *TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492*
  - *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*
  - *Other - ECDHE-RSA-AES128-GCM-SHA256*
  - *Other - ECDHE-RSA-AES256-GCM-SHA384*

  ]

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

FCS_TLSS_EXT.1.3 The TSF shall [*perform RSA key establishment with key size [selection: 2048 bits, 3072 bits]; generate EC Diffie-Hellman parameters over NIST curves [secp256r1, secp384r1] and no other curves; generate Diffie-Hellman parameters of size [2048 bits]*].

5.2.3 Class: Identification and Authentication (FIA)

FIA_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 shall support passwords of 15 characters or greater;*

FIA_UIA_EXT.1 User Identification and Authentication

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

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

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

FIA_UAU_EXT.2 Password-based Authentication Mechanism

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

FIA_UAU.7 Protected Authentication Feedback

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

FIA_X509_EXT.1 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 Section 5*].
- The TSF shall validate the extendedKeyUsage field according to the following rules:
    - *Certificates used for trusted updates and executable code integrity verification shall have the Code Signing purpose (id-kp 3 with OID 1.3.6.1.5.5.7.3.3) in the extendedKeyUsage field.*
    - *Server certificates presented for TLS shall have the Server Authentication purpose (id-kp 1 with OID 1.3.6.1.5.5.7.3.1) in the extendedKeyUsage field.*
    - *Client certificates presented for TLS shall have the Client Authentication purpose (id-kp 2 with OID 1.3.6.1.5.5.7.3.2) in the extendedKeyUsage field.*
    - *OCSP certificates presented for OCSP responses shall have the OCSP Signing purpose (id-kp 9 with OID 1.3.6.1.5.5.7.3.9) in the extendedKeyUsage field.*

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

FIA_X509_EXT.2 X.509 Certificate Authentication

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

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

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 Name, Country, [State, Locality, Email Address, Challenge Password, and Optional Company Name]*].

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

5.2.4 Class: Security Management (FMT)

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

FMT_MTD.1 Management of TSF Data
FMT_MTD.1.1 The TSF shall restrict the ability to *manage* the *TSF data* to the *Security Administrators*

FMT_SMF.1 Specification of Management Functions
FMT_SMF.1.1 The TSF shall be capable of performing the following management functions

- *Ability to administer the TOE locally and remotely;*
- *Ability to configure the access banner;*
- *Ability to configure the session inactivity time before session termination or locking;*
- *Ability to update the TOE, and to verify the updates using [digital signature] capability prior to installing those updates;*
- *[*
    - *Ability to configure the cryptographic functionality]*

FMT_SMR.2 Restrictions on Security Roles
FMT_SMR.2.1 The TSF shall maintain the roles:

- *Security Administrator*

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

FMT_SMR.2.3 The TSF shall ensure that the conditions

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

are satisfied.

5.2.5 Class: Protection of the TSF (FPT)

FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)
FPT_SKP_EXT.1.1 The TSF shall prevent reading of all pre-shared keys, symmetric keys and private keys.

FPT_APW_EXT.1 Protection of Administrator Passwords
FPT_APW_EXT.1.1 The TSF shall store passwords in non-plaintext form.
FPT_APW_EXT.1.2 The TSF shall prevent the reading of plaintext passwords.

FPT_STM.1 Reliable Time Stamps
FPT_STM.1.1 The TSF shall be able to provide reliable time stamps.

FPT_TUD_EXT.1 Trusted Update
FPT_TUD_EXT.1.1 The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software and [*the most recently installed version of the TOE firmware/software*].

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

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

FPT_TST_EXT.1: TSF Testing
FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests [*during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [manual reboot]*] to demonstrate the correct operation of the TSF: [*AES Known Answer Test, HMAC Known Answer Test, RNG/DRBG Known Answer Test, SHA Known Answer Test, RSA Signature Known Answer Test (both signature/verification), DH Known Answer Test, ECDH Known Answer Test*].

5.2.6 Class: TOE Access (FTA)

FTA_SSL_EXT.1 TSF-initiated Session Locking
FTA_SSL_EXT.1.1 The TSF shall, for local interactive sessions, [

- *terminate the session]*

after a Security **Administrator**-specified time period of inactivity.

FTA_SSL.3 TSF-initiated Termination
FTA_SSL.3.1 **Refinement:** The TSF shall terminate **a remote** interactive session after a *Security Administrator-configurable time interval of session inactivity*.

FTA_SSL.4 User-initiated Termination
FTA_SSL.4.1 **Refinement:** The TSF shall allow **Administrator**-initiated termination of the **Administrator**'s own interactive session.

FTA_TAB.1 Default TOE Access Banners
FTA_TAB.1.1 **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**.

5.2.7 Class: Trusted Path/Channels (FTP)

FTP_ITC.1 Inter-TSF trusted channel

FTP_ITC.1.1 The TSF shall be **capable of using [*TLS*] to** provide a trusted communication channel between itself and **authorized IT entities supporting the following capabilities: audit server** that is logically distinct from other communication channels and provides assured identification of its end points and protection of the channel data from disclosure and detection of modification of the channel data.

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

FTP_ITC.1.3 The TSF shall initiate communication via the trusted channel for [*secure audit log transfer*].

FTP_TRP.1 Trusted Path

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

FTP_TRP.1.2 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**.

5.3 TOE SFR Dependencies Rationale for SFRs

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

5.4 Security Assurance Requirements

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

Table 6 Security Assurance Requirements

Assurance Class	Components	Components Description
Security Target	ASE_CCL.1	Conformance Claims
	ASE_ECD.1	Extended Components Definition
	ASE_INT.1	ST Introduction
	ASE_OBJ.1	Security Objectives for the Operational Environment
	ASE_REQ.1	Stated Security Requirements
	ASE_SPD.1	Security Problem Definition
	ASE_TSS.1	TOE Summary Specification
Development	ADV_FSP.1	Basic Functional Specification
Guidance Documents	AGD_OPE.1	Operational User Guidance
	AGD_PRE.1	Preparative Procedures
Life Cycle Support	ALC_CMC.1	Labeling of the TOE
	ALC_CMS.1	TOE CM Coverage
Tests	ATE_IND.1	Independent Testing – Sample
Vulnerability Assessment	AVA_VAN.1	Vulnerability Survey

6. TOE Summary Specification

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

Table 7 TOE Summary Specification SFR Description

#	TOE SFR	Rationale
1	FAU_GEN.1	The TOE provides extensive auditing capabilities. The TOE generates a comprehensive set of audit logs that identify specific TOE operations. For each event, the TOE records the date and time of each event, the type of event, the subject identity, and the outcome of the event. The types of events that cause audit records to be generated include identification and authentication related events, and administrative events. Each of the events is specified in the audit record is in enough detail to identify the user for which the event is associated (e.g. user identity, MAC address, IP address), when the event occurred, where the event occurred, the outcome of the event, and the type of event that occurred. The logs for all the appliances can be viewed via the remote GUI interface or through the CLI (local or remote). Additionally, the TOE supports remote audit logging using the syslog standard with an external server. Audit messages are entered into the log and the subset of the log contents are sent to the syslog server. When an administrative command is executed, the TOE sets up the session data structure which includes the “user identity”. When an audit log is generated, the session data is passed along with the audit information and the TOE extracts the “user identity” from the session data structure. The TOE generates the following types of audit logs during operation: • Start-up of the TOE from both cold boot and reboot, • Shutdown of the TOE (when shut down from the local CLI, Remote CLI, and GUI), • All administrative actions (both security relevant and non-security relevant) from the local CLI, Remote CLI, and GUI, • Remote administrative HTTPS/TLS connection establishment, • Remote administrative HTTPS/TLS connection closure, • Errors during Remote administrative HTTPS/TLS connection establishment, • Remote administrative SSH connection establishment, • Remote administrative SSH connection closure, • Errors during Remote administrative SSH connection establishment, • Generation of self-signed certificates, • Import of certificates, • Deletion of certificates, • Successful authentication attempts (from the local CLI, Remote CLI, and GUI), • Unsuccessful authentication attempts (from the local CLI, Remote CLI, and GUI), • Unsuccessful certificate validation for the presence of the basicConstraints extension missing, • Unsuccessful certificate validation for the CA flag is set to TRUE for all CA certificates, • Unsuccessful certificate validation for trust chain verification failure, • Unsuccessful certificate validation for revocation status, • All attempts to update the TOE software, • Changes to time, • Start of a local administrative session, • End of a local administrative session, • Administration session timeout (from the local CLI, Remote CLI, and GUI).
2	FAU_GEN.2	The TOE ensures that each auditable event is associated with the user that triggered the event. For example, a human user, user identity or related session ID would be included in the audit record. For an IT entity or device, the IPv4 address, MAC address, host name, or other configured identification is included in the audit record. The audit record is generated with the required information and stored plaintext on the device.
3	FAU_STG_EXT.1	The TOE is configured to export syslog records to a specified, external syslog server. The TOE protects communications with an external syslog server via mutually authenticated TLS. When communicating with an external syslog server, the TOE acts as a TLS client. The TOE then periodically initiates a connection with the syslog server. Once the server has accepted the TLS connection as a TLS server, the TOE pushes the audit logs to the syslog server over the secure channel.

#	TOE SFR	Rationale
		The maximum size of audit records stored by the TOE can be configured by an administrator. The upper limit on local audit storage is based on the amount of available hard drive space, but an administrator can set a lower limit if desired. For audit records stored internally to the TOE the audit records are stored in a circular log file where the TOE overwrites the oldest audit records when the audit trail becomes full. When storage capacity is reached, the oldest of the six audit files is overwritten. Only Authorized Administrators can clear the local logs, and local audit records are stored in a directory that does not allow administrators to modify the contents. However, the Authorized Administrator may do a onetime configuration that will not allow the administrator to erase logs. This command is irreversible and does not reset even if the machine is returned to factory defaults.
4	FCS_CKM.1	The TOE can create a RSA public-private key pair with a RSA key size of 2048 and 4096 bits. The RSA algorithm implementation is provided by the SA cryptographic library. The RSA key pair can be used to generate a Certificate Signing Request (CSR). The TOE is fully compliant to both SP 800-56A and SP 800-56B. The TOE implements each “shall” statement in each standard and does not implement any “shall not” statements in either of the standards. The RSA, FFC and ECDHE schemes are used for HTTPS/TLS Server and TLS Client communications. For SSH, the TOE uses the ECDH scheme.
5	FCS_CKM.2	In support of secure cryptographic protocols, the TOE supports key establishment schemes, including, • FFC Diffie-Hellman as specified in NIST SP 800-56A, • Elliptic Curve Diffie-Hellman as specified in NIST SP 800-56A, • RSA Key Establishment as specified in SP NIST 800-56B. The TOE is fully compliant to both SP 800-56A and SP 800-56B. The TOE implements each “shall” statement in each standard and do not implement any “shall not” statements in either of the standards. The FFC and ECDHE schemes are used for HTTPS/TLS Server and TLS Client communications. The RSA scheme is used for HTTPS/TLS Server and TLS Client communications. For TLS, the TOE acts as both a sender and receiver. For SSH, the TOE acts only as a receiver and uses the ECDH scheme. In the instance where a decryption error occurs, the TOE does not reveal the particular error that occurred, in accordance with NIST SP 800-56B.
6	FCS_CKM.4	The TOE meets all requirements specified in FIPS 140-2 for destruction of keys and Critical Security Parameters (CSPs) when no longer required for use. The TOE stores several types of keys in volatile memory in plaintext, including, • Diffie-Hellman Private/Public Key Pair, • Elliptic Curve Diffie-Hellman Private/Public Key Pair, • SSH Session Encryption Key, • SSH Session Integrity Key, • TLS Session Encryption Key, • TLS Session Integrity Key. Each plaintext key stored in volatile memory is associated with a cryptographic session. In each instance, after the session closes, the key is overwritten with the value “00” After the overwrite operation is complete, the TOE performs a specific "read-verify" operation to confirm that the storage space no longer contains the key. The TOE stores RSA key pairs used for TLS and SSH in non-volatile storage. These are overwritten three times using a random pattern provided by the SP 800-90A DRBG.
7	FCS_COP.1(1)	The TOE provides symmetric encryption and decryption capabilities using AES in CBC and GCM mode (128, 256 bits) as described AES as specified in ISO 18033-3. AES is implemented in support of the following protocols: TLS, and SSH.
8	FCS_COP.1(2)	The TOE provides cryptographic signature services using RSA Digital Signature Algorithm with key sizes of 2048, 3072 and 4096 bits as specified in FIPS PUB 186-4, “Digital Signature Standard”.
9	FCS_COP.1(3)	The TOE provides cryptographic hashing services using SHA-1 as specified in FIPS Pub 180-3 “Secure Hash Standard.” SHS hashing is used within several services including, hashing, TLS/HTTPS, and SSH. SHA-512 is used in conjunction with RSA signatures 4096 for verification of software image integrity. SHA-256 and SHA-512 are used in conjunction with SSH session establishment and SHA-256 and SHA-384

#	TOE SFR	Rationale
		are used in conjunction with TLS session establishment as part of the ciphers used during the handshake process.
10	FCS_COP.1(4)	The TOE provides keyed-hashing message authentication services using HMAC-SHA-1, HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512. The product supports the following cryptographic parameters for MACing, as specified in ISO/IEC 9797-2:2011: • Key length: 160-bits, 256-bits, 384-bits, 512-bits • Hash function used: SHA-1, SHA-256, SHA-384, and SHA-512 • Block size: 512-bits, 1024-bits • Output MAC: 160-bits, 256-bits, 384-bits, 512-bits
11	FCS_RBG_EXT.1	The TOE implements a NIST-approved HMAC_DRBG, as specified in SP 800-90. The TOE implements a random bit generator in support of various cryptographic operations, including, RSA key establishment schemes, Diff-Hellman key establishment schemes, TLS session establishment and SSH session establishment. The entropy source used to seed the Deterministic Random Bit Generator (e.g. based on SP 800-90A/B/C) is a random set of bits or bytes that are regularly supplied to the DRBG by polling four different set of software sources in threads. All entropy is continuously health tested by the DRBG as per the tests defined in section 11.3 of SP 900-90A before being used as a seed. Any initialization or system errors during bring-up or processing of this system causes a reboot resulting in the DRBG being reseeded.
13	FCS_TLSC_EXT.2	In support of secure communication with external entities, the TOE supports the TLS protocol acting as a TLS client. TLS is used to facilitate communication with the following entities, • Syslog Servers In support of these connections, the TOE support TLS 1.1 and TLS 1.2. No other TLS protocol versions, such as, TLS 1.0 or SSL 3.0 are offered. The following cipher suites are supported for communications with syslog servers: • TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268 • 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_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492 • 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 • Other - ECDHE-RSA-AES128-GCM-SHA256 • Other - ECDHE-RSA-AES256-GCM-SHA384 The TOE supports full validation of the presented TLS server certificates, including, Common Name, DNS Name, URI Name, and Service Name. Additionally, the TOE presents its client certificate to the syslog server for validation. Reference Identifier support for IP addresses and wildcards are present in the evaluated configuration. The following elliptic curve extensions are supported by the TOE: • secp256r1 • secp384r1 • secp521r1 All other elliptic curve extensions are denied. Certificate pinning is not supported by the TOE.
14	FCS_HTTPS_EXT.1	In support of secure communication with external entities, the TOE supports the TLS protocol acting as a TLS server. TLS is used to facilitate communication with the following entities, • Remote administrators

#	TOE SFR	Rationale
		The communication with remote administrators is over a TLS-protected HTTPS connection. In support of these connections, the TOE supports TLS 1.1 and TLS 1.2. Connections using another version of TLS or SSL, such as, TLS 1.0 or SSL 3.0 are actively denied by the TOE. The following cipher suites are supported for communications with remote administrators: • TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268 • 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_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492 • 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 • Other - ECDHE-RSA-AES128-GCM-SHA256 • Other - ECDHE-RSA-AES256-GCM-SHA384 All other proposed cipher suites are denied. 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.
15	FCS_TLSS_EXT.1	In support of secure communication with external entities, the TOE supports the TLS protocol acting as a TLS server. TLS is used to facilitate communication with the following entities, • Remote administrators The communication with remote administrators is over a TLS-protected HTTPS connection. The TOE supports key agreement parameters including DH Group modulus, certificates, TLS protocol version, key agreement algorithm, and data integrity algorithm. In support of these connections, the TOE supports TLS 1.1 and TLS 1.2. Connections using another version of TLS or SSL, such as, TLS 1.0 or SSL 3.0 are actively denied by the TOE. The following cipher suites are supported for communications with remote administrators: • TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268 • 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_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492 • TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492 • 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 • Other - ECDHE-RSA-AES128-GCM-SHA256 • Other - ECDHE-RSA-AES256-GCM-SHA384 All other proposed cipher suites are denied.
16	FCS_SSHS_EXT.1	The TOE uses SSH for to facilitate secure remote administrative sessions (CLI). The TOE's SSH implementation supports the following, • Use of 2048-bit RSA keys in support of SSH_RSA for public key-based authentication; • Dropping SSH packets greater than 1522 bytes. This is accomplished by buffering all data for a particular SSH packet transmission until the buffer limit is reached and then dropping the packet; • Strict compliance with RFCs 4251, 4252, 4253, and 4254, ○ No optional options included in the RFCs have been implemented; • Encryption algorithms aes128-cbc, aes256-cbc, AEAD_AES_128_GCM, and AEAD_AES_256_GCM to ensure confidentiality of the session;

#	TOE SFR	Rationale
		• Password based authentication; • Public key based authentication; • Hashing algorithm HMAC-SHA2-256 or HMAC-SHA2-512 to ensure the integrity of the session; • Enforcement of ecdh-sha2-nistp256, ecdh-sha2-nistp384, and ecdh-sha2-nistp521 for key exchange algorithms. During an SSH session, the TOE monitors the connection length in transmitted data and time and performs a rekey no longer than every hour or prior to 1 gigabyte of transmitted data.
17	FIA_PMG_EXT.1	The TOE supports the local definition of users with corresponding passwords. The passwords can be composed of any combination of upper and lower case letters, numbers, and special characters (that include: “!”, “@”, “#”, “$”, “%”, “^”, “&”, “*”, “(“, “)”, “”, “+”, “-“, “=”, “.”, “/”, “\”, “:”, “;”, “<”, “>”, “[“, “]”, “_”, “{“, “}”, “|”, “~” “`” . The minimum password length is settable by the Authorized Administrator. When the TOE is configured for "Common Criteria Compliance" the minimum password length is set to 15 characters.
18	FIA_UIA_EXT.1	The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed. Administrative access to the TOE is facilitated through one of several interfaces, • Directly connecting to the TOE • Remotely connecting via SSHv2 • Remotely connecting to the GUI via HTTPS/TLS Regardless of the interface at which the administrator interacts, the TOE will enforce username and password authentication or public-key based authentication. 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. The process for authentication is the same for administrative access whether administration is occurring via direct connection or remotely. At initial login, the administrative user is prompted to provide a username. After the user provides the username, the user is prompted to provide the administrative password associated with the user account. The TOE then either grant administrative access (if the combination of username and password is correct) or indicate that the login was unsuccessful. The TOE does not provide a reason for failure in the cases of a login failure.
18 19	FIA_UAU_EXT.2 FIA_UAU.7	The TOE requires all users to be successfully identified and authenticated before allowing any TSF mediated actions to be performed. Administrative access to the TOE is facilitated through one of several interfaces, • Directly connecting to the TOE • Remotely connecting via SSHv2 • Remotely connecting to the GUI via HTTPS/TLS Regardless of the interface at which the administrator interacts, the TOE will enforce username and password authentication or public-key based authentication (for SSH connections only). 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. The process for authentication is the same for administrative access whether administration is occurring via direct connection or remotely. At initial login, the administrative user is prompted to provide a username. After the user provides the username, the user is prompted to provide the administrative password associated with the user account. The TOE then either grant administrative access (if the combination of username and password is correct) or indicate that the login was unsuccessful. The TOE does not provide a reason for failure in the cases of a login failure. For all authentication, regardless of the interface, the TOE displays only "*" characters when the administrative password is entered so that the password is obscured.
20	FIA_X509_EXT.1 FIA_X509_EXT.2	

#	TOE SFR	Rationale
	FIA_X509_EXT.3	The TOE uses X.509v3 certificates as defined by RFC 5280 to support the following connections, • TLS connections with external syslog servers. The TOE creates Certificate Signing Request (CSRs). These signing requests contain the following fields, • Public Key • Common Name • Country This signing request is then sent to a CA for generation of a CA signed certificate. The TOE supports the following methods to obtain a certificate from a CA: • Simple Certificate Enrollment Protocol (SCEP) Each local certificate is digitally signed providing protection from unauthorized modification. If a certificate is modified in any way, it would be invalidated and rendered useless. 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. Each CA corresponds to a trust point. The X.509 certificates are validated using the certificate path validation algorithm defined in RFC 5280, which can be summarized as follows: • The public key algorithm and parameters are checked • The current date/time is checked against the validity period revocation status is checked • Issuer name of X matches the subject name of X+1 • Name constraints are checked • Policy OIDs are checked • Policy constraints are checked; issuers are ensured to have CA signing bits • Path length is checked • Critical extensions are processed In order to verify the revocation status of the presented certificates, a Certificate Revocation List (CRL) is used. The physical security of the TOE (A.PHYSICAL_PROTECTION) protects the TOE 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. If the connection to determine the certificate validity cannot be established, the TOE accepts the certificate. The TOE uses local clock (which may be synced with an NTP server) for validation of the validity period as time mechanism. The TOE is configured with a single certificate for communication. A new certificate can be uploaded VIA the UI, overwriting the previous certificate. All communication from the TOE will select the configured certificate for communication.
21	FMT_MOF.1(1)/ Trusted Update	The TOE does not provide automatic updates to the software version running on the TOE. The Security Administrators (a.k.a. Authorized Administrators) can query the software version running on the TOE, and can initiate updates to (replacements of) software images. When software updates are made available, the Authorized Administrators can obtain, verify the integrity of, and install those updates. This verification uses digital signatures.
22	FMT_MTD.1	The TOE provides the ability for Security Administrators (a.k.a Authorized Administrators) to access TOE data, such as audit data, configuration data, security attributes, session thresholds and updates. Access to this data is governed by the privileges assigned to the administrative users. None of this functionality is accessible prior to the administrator logging into the TOE. The term “Authorized Administrator” is used in this ST to refer to any of the predefined user privilege levels.
23	FMT_SMF.1	The TOE provides all the capabilities necessary to securely manage the TOE. The Security Administrators (a.k.a. Authorized Administrators) user can connect to the TOE using the CLI to perform these functions via remote CLI over SSHv2, at the local console, or via remote GUI over an HTTPS connection. The specific management capabilities available from the TOE include:

#	TOE SFR	Rationale
		• Local and remote administration of the TOE and the services provided by the TOE via the TOE CLI/GUI, as described above, • Ability to configure the access banner, • Ability to configure the session inactivity time before session termination or locking, • Ability to update the TOE, and to verify the updates using digital signature capability prior to installing those updates, The ability to manage the cryptographic functionality which allows the Authorized Administrator the ability to identify and configure the algorithms used to provide protection of the data, such as generating RSA keys.
24	FMT_SMR.2	The TOE supports multiple administrative roles when accessing the administrative interface through the local or remote CLI. These roles define the access that is allowed per role. The pre-defined Admin and Auditor groups collectively correspond to the Security Administrator role when the CLI is used. Additionally, the TOE supports one authorized user role when accessing the TOE through the remote GUI interface. This role has full access to the TOE management capabilities defined in the NDcPP. This entity is referred to as the Security Administrator group in the guidance documentation and corresponds to the Security Administrator role when the GUI is used.
25	FPT_SKP_EXT.1	All keys stored on the TOE are protected from unauthorized modification and substitution. The TOE stores symmetric keys only in volatile memory never on persistent media. The TOE admin interface does not provide any mechanism to view sensitive data (passwords or keys) once stored. Unauthenticated operators do not have write access to modify, change, or delete keys. The TOE stores all asymmetric keys in a secure directory that is not readily accessible to administrators; therefore, there is no administrative interface access provided to directly manipulate the keys.
26	FPT_APW_EXT.1	No passwords are ever stored as clear text. Passwords are stored on the TOE in a secured partition in non-plaintext. Prior to writing on disks each password is hashed (SHA-256) using the PBKDF2 algorithm. During subsequent authentication attempts passwords entered are converted using the same PBKDF2 algorithm. This is compared to the digest value for that user stored in the secured partition. Access is only granted if the values match.
27	FPT_TST_EXT.1	The TOE runs a suite of self-tests during initial start-up to verify its correct operation. If any of the tests fail, the Authorized Administrator must connect to the device via the local interface and view the results of the self-tests as they are executed. During the system bootup process (power on or reboot), the TOE performs various power-on self-test (POSTs) for the cryptographic components of the TOE. During initialization and self-test execution, the module inhibits all access to the cryptographic algorithms. Additionally, the power-on self-tests are performed after the cryptographic systems are initialized but prior to the underlying OS initialization of external interfaces; this prevents the security appliances from passing any data before completing self-tests. In the event of a power-on self-test failure, the cryptographic module will force the platform to reload and reinitialize the operating system and cryptographic components. This operation ensures no cryptographic algorithms can be accessed unless all power on self-tests are successful. These tests include: • AES Known Answer Test - For the encrypt test, a known key is used to encrypt a known plain text value resulting in an encrypted value. This encrypted value is compared to a known encrypted value to ensure that the encrypt operation is working correctly. The decrypt test is just the opposite. In this test a known key is used to decrypt a known encrypted value. The resulting plaintext value is compared to a known plaintext value to ensure that the decrypt operation is working correctly. • HMAC Known Answer Test - For each of the hash values listed, the HMAC implementation is fed known plaintext data and a known key. These values are used to generate a MAC. This MAC is compared to a known MAC to verify that the HMAC and hash operations are operating correctly. • RNG/DRBG Known Answer Test - For this test, known seed values are provided to the DRBG implementation. The DRBG uses these values to generate random bits. These random bits are compared to known random bits to ensure that the DRBG is operating correctly. • SHA Known Answer Test – For each of the values listed, the SHA implementation is fed known data and key. These values are used to generate a hash. This hash is compared to a known value to verify they match and the hash operations are operating correctly. • RSA Signature Known Answer Test (both signature/verification) - This test takes a known plaintext value and Private/Public key pair and used the public key to encrypt the data. This value is compared to a known encrypted value to verify that encrypt operation is working

#	TOE SFR	Rationale
		properly. The encrypted data is then decrypted using the private key. This value is compared to the original plaintext value to ensure the decrypt operation is working properly. • DH Known Answer Test – This test takes known input to the "z" calculation for Diffie-Hellman and compares the result to a known "z" value. • ECDH Known Answer Test – This test takes known input to the "z" calculation for Elliptic Curve Diffie-Hellman and compares the result to a known "z" value. Software Integrity Test - This test is run automatically whenever the system images are loaded and confirms through use of digital signature verification that the image file that's about to be loaded was properly signed and maintained its integrity since being signed. The system image is digitally signed prior to being made available for download from Bluecoat/Symantec.
	FPT_TUD_EXT.1	Authorized Administrator can query the software version running on the TOE, and can initiate updates to software images. When software updates are made available, an administrator can obtain, verify the integrity of via digital signature, and install those updates. The updates can be downloaded from Upgrades.soleranetworks.com. The TOE image files are digitally signed so their integrity can be verified during the boot process, and an image that fails an integrity check will not be loaded. The public keys used by the update verification mechanism are contained on the TOE. As part of the build process, the update image is signed with the Bluecoat/Symantec private key. This is done using an RSA 4096/SHA-512 digital signature. Only if the signature/hash is correct, will the image be installed. If an update is unsuccessful, a message is delivered to the user. Since the update process attempts to update a different copy than what is currently being run, the current active image remains the same and the user continues to run the same code that was being run before the upgrade attempt was made.
	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.
	FTA_SSL_EXT.1	The TOE provides the administrative user to defined inactivity time out periods for administrative sessions. The inactivity period for CLI (local and remote) and GUI (remote) administrative access are maintained separately and are configured separately through the TOE administrative interfaces. If an administrative session remains inactive for the configured length of time, the administrative session is terminated. After termination, administrative authentication is required to access any of the administrative functionality of the TOE. This is applicable from both local and remote administrative sessions.
	FTA_SSL.3 FTA_SSL.4	The TOE provides the administrative user to defined inactivity time out periods for administrative sessions. The inactivity period for CLI (local and remote) and GUI (remote) administrative access are maintained separately and are configured separately through the TOE administrative interfaces. If an administrative session remains inactive for the configured length of time, the administrative session is terminated. After termination, administrative authentication is required to access any of the administrative functionality of the TOE. This is applicable from both local and remote administrative sessions. An Authorized Administrator can exit out of both local and remote administrative sessions. When accessing the TOE via the CLI (both local and remote), the exit command is used. When accessing the TOE via the remote GUI, the logout button is used.
	FTA_TAB.1	For TOE administration, the GUI (TLS/HTTPS), CLI (SSH) and local console CLI are available. Prior to an administrative user authenticating, that user is presented with an access display banner which displays an advisory notice and consent warning message regarding unauthorized use of the TOE. This banner will be displayed prior to allowing Administrator access through those interfaces.
	FTP_ITC.1	The TOE protects communications with authorized IT entities, as follows: • Trusted channels with audit servers are protected via TLS. This protects the data from disclosure by encryption and by checksums that verify that data has not been modified.
	FTP_TRP.1	All remote administrative communications take place over a secure encrypted session. Remote CLI connections take place over an SSHv2 tunnel. The SSHv2 session is encrypted using AES encryption. Remote GUI connections take place over a TLS/HTTPS connection. The TLS session is encrypted using AES encryption. The remote administrators are able to initiate both SSHv2 and TLS/HTTPS communications with the TOE. The TOE rejects all insecure remote authentication attempts (e.g., telnet and HTTP).

7. Acronyms

This section describes the acronyms used throughout this document.

Table 8 Acronyms

Acronym	Definition
CA	Certificate Authority
CC	Common Criteria
CLI	Command Line Interface
CRL	Certificate Revocation List
DH	Diffie-Hellman
ECDH	Elliptic Curve Diffie-Hellman
GUI	Graphical User Interface
OS	Operating System
POST	Power On Self-Test
PP	Protection Profile
SA	Security Analytics
SHS	Secure Hashing Standard
SSH	Secure Shell
TLS	Transport Layer Security

8. Extended Component Definitions

The Security Functional Components found in Table 9 below as well as in Section 5 of this Security Target are taken directly from the Collaborative Protection Profile for Network Devices, Version 1.0. These components claim exact conformance to the definitions within the PP and, as identified in Section 5.1 of the PP, “are identified by having a label ‘_EXT’ at the end of the SFR name.”

Table 9 Extended Components

Class	Family/Component	Description
FAU: Security Audit	FAU_STG_EXT.1	Protected Audit Event Storage
FCS: Cryptographic Support	FCS_HTTPS_EXT.1	HTTPS Protocol
	FCS_SSHS_EXT.1	SSH Server Protocol
	FCS_TLSC_EXT.2	TLS Client Protocol with Authentication
	FCS_TLSS_EXT.1	TLS Server Protocol
	FCS_RBG_EXT.1	Random Bit Generation
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_X509_EXT.1	Certificate Validation
	FIA_X509_EXT.2	Certificate Authentication
	FIA_X509_EXT.3	Certificate Requests
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 TUD EXT.1	Trusted Update
	FPT TST EXT.1	TSF Testing
FTA: TOE Access	FTA_SSL_EXT.1	TSF-initiated Session Locking

8.1 FAU_STG_EXT.1 Protected Audit Event Storage

FAU_STG_EXT.1 Protected audit event storage requires the TSF to use a trusted channel implementing a secure protocol.

Management: FAU_STG_EXT.1
The following actions could be considered for the management functions in FMT:
a) The TSF shall have the ability to configure the cryptographic functionality.

Audit: FAU_STG_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) No audit necessary.

Hierarchical to: No other components.
Dependencies: FAU_GEN.1 Audit data generation
FTP_ITC.1 Inter-TSF Trusted Channel

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

Application Note 94
For selecting the option of transmission of generated audit data to an external IT entity the TOE relies on a non-TOE audit server for storage and review of audit records. The storage of these audit records and the ability to allow the administrator to review these audit records is provided by the operational environment in that case.

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 *[selection: drop new audit data, overwrite previous audit records according to the following rule: [assignment: rule for overwriting previous audit records], [assignment: other action]]* when the local storage space for audit data is full.

Application Note 95
The external log server might be used as alternative storage space in case the local storage space is full. The 'other action' could in this case be defined as 'send the new audit data to an external IT entity'.

8.2 FCS_RBG_EXT.1 Random Bit Generation

FCS_RBG_EXT.1 Random Bit Generation requires random bit generation to be performed in accordance with selected standards and seeded by an entropy source.

Management: FCS_RBG_EXT.1
The following actions could be considered for the management functions in FMT:
a) There are no management activities foreseen

Audit: FCS_RBG_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: failure of the randomization process

Hierarchical to: No other components
Dependencies: No other components

FCS_RBG_EXT.1.1 The TSF shall perform all deterministic random bit generation services in accordance with ISO/IEC 18031:2011 using *[selection: 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 *[selection: [assignment: number of software-based sources] software-based noise source, [assignment: number of hardware-based sources] hardware-based noise source]* with minimum of *[selection; 128 bits, 192 bits, 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.

8.3 FCS_HTTPS_EXT.1 HTTPS Protocol

FCS_HTTPS_EXT.1 HTTPS requires that HTTPS be implemented according to RFC 2818 and supports TLS.

Management: FCS_HTTPS_EXT.1
The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_HTTPS_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) There are no auditable events foreseen

Hierarchical to: No other components
Dependencies: FCS_TLS_EXT.1 TLS Protocol

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

FCS_HTTPS_EXT.1.2 The TSF shall implement the HTTPS protocol using TLS.

FCS_HTTPS_EXT.1.3 The TSF shall establish the connection only if *[selection: the peer presents a valid certificate during handshake, the peer initiates handshake]*.

8.4 FCS_SSHS_EXT.1 SSH Server Protocol

FCS_SSHS_EXT.1 SSH Server requires that the server side of SSH be implemented as specified.

Management: FCS_SSHS_EXT.1
The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_SSHS_EXT.1

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

a) Failure of SSH session establishment.
b) SSH session establishment
c) SSH session termination

Hierarchical to: No other components
Dependencies: FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (Signature Verification)
FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)

FCS_SSHS_EXT.1.1 The TSF shall implement the SSH protocol that complies with RFCs 4251, 4252, 4253, 4254, and *[selection: 5647, 5656, 6187, 6668, no other RFCs]*.

Application Note 109
The ST author selects which of the additional RFCs to which conformance is being claimed. Note that these need to be consistent with selections in later elements of this component (e.g., cryptographic algorithms permitted). RFC 4253 indicates that certain cryptographic algorithms are "REQUIRED". This means that the implementation must include support, not that the algorithms must be enabled for use. Ensuring that algorithms indicated as "REQUIRED" but not listed in the later elements of this component are implemented is out of scope of the assurance activity for this requirement.

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 *[assignment: number of bytes]* bytes in an SSH transport connection are dropped.

Application Note 110
RFC 4253 provides for the acceptance of "large packets" with the caveat that the packets should be of "reasonable length" or dropped. The assignment should be filled in by the ST author with the maximum packet size accepted, thus defining "reasonable length" for the TOE.

FCS_SSHS_EXT.1.4 The TSF shall ensure that the SSH transport implementation uses the following encryption algorithms and rejects all other encryption algorithms*: [selection: aes128-cbc, aes256-cbc, AEAD_AES_128_GCM, AEAD_AES_256_GCM].*

FCS_SSHS_EXT.1.5 The TSF shall ensure that the SSH transport implementation *uses [assignment: List of public key algorithms]* 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 *[assignment: List of MAC algorithms]* as its MAC algorithm(s) and rejects all other MAC algorithm(s).

FCS_SSHS_EXT.1.7 The TSF shall ensure that *[assignment: List of key exchange 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.

8.5 FCS_TLSC_EXT.2 TLS Client Protocol with Authentication

FCS_TLSC_EXT.2 TLS Client requires that the client side of the TLS implementation include mutual authentication.

Management: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2
The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_TLSC_EXT.1, FCS_TLSC_EXT.2
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of TLS session establishment.
b) TLS session establishment
c) TLS session termination

Hierarchical to: FCS_TLSC_EXT.1 TLS Client Protocol
Dependencies: FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (Signature Verification)
FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSC_EXT.2.1 The TSF shall implement [selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)] supporting the following ciphersuites:

[selection:
TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
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_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
TLS_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
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289].

Application Note 115
The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the ciphersuites that are supported. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. Note that RFC 5246 makes TLS_RSA_WITH_AES_128_CBC_SHA a mandatory ciphersuite, but it is treated as optional for the purposes of conformance with this cPP (i.e. the selection of 'TLS 1.2 (RFC 5246)' will be accepted as

conformant with this SFR even if TLS_RSA_WITH_AES_128_CBC_SHA is not one of the ciphersuites listed in the ST).

FCS_TLSC_EXT.2.2 The TSF shall verify that the presented identifier matches the reference identifier according to RFC 6125.

Application Note 116
The rules for verification of identify are described in Section 6 of RFC 6125. The reference identifier is established by the user (e.g. entering a URL into a web browser or clicking a link), by configuration (e.g. configuring the name of a mail server or authentication server), or by an application (e.g. a parameter of an API) depending on the application service. Based on a singular reference identifier's source domain and application service type (e.g. HTTP, SIP, LDAP), the client establishes all reference identifiers which are acceptable, such as a Common Name for the Subject Name field of the certificate and a (case-insensitive) DNS name, URI name, and Service Name for the Subject Alternative Name field. The client then compares this list of all acceptable reference identifiers to the presented identifiers in the TLS server's certificate.

FCS_TLSC_EXT.2.3 The TSF shall [*selection: perform RSA key establishment with key size [selection: 2048 bits, 3072 bits, 4096 bits*]; generate EC Diffie-Hellman parameters over NIST curves [*selection: secp256r1, secp384r1, secp521r1*] and no other curves; generate Diffie-Hellman parameters of size [*selection: 2048, bits, 3072 bits*]].

Application Note 117
Validity is determined by the identifier verification, certificate path, the expiration date, and the revocation status in accordance with RFC 5280.

FCS_TLSC_EXT.2.4 The TSF shall present the Supported Elliptic Curves Extension in the Client Hello with the following NIST curves: *[assignment: List of supported curves including an option for 'none'].*

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

Application Note 118
The use of X.509v3 certificates for TLS is addressed in FIA_X509_EXT.2.1. This requirement adds that this use must include the client must be capable of presenting a certificate to a TLS server for TLS mutual authentication.

8.6 FCS_TLSS_EXT.1 TLS Server Protocol

FCS_TLSS_EXT.1 TLS Server requires that the server side of TLS be implemented as specified.

Management: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2
The following actions could be considered for the management functions in FMT:

a) There are no management activities foreseen.

Audit: FCS_TLSS_EXT.1, FCS_TLSS_EXT.2
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Failure of TLS session establishment.
b) TLS session establishment
c) TLS session termination

Hierarchical to: No other components
Dependencies: FCS_CKM.1 Cryptographic Key Generation

FCS_COP.1(1) Cryptographic operation (AES Data encryption/decryption)
FCS_COP.1(2) Cryptographic operation (Signature Verification)
FCS_COP.1(3) Cryptographic Operation (Hash Algorithm)
FCS_RBG_EXT.1 Random Bit Generation

FCS_TLSS_EXT.1.1 The TSF shall implement *[selection: TLS 1.2 (RFC 5246), TLS 1.1 (RFC 4346)]* supporting the following ciphersuites:

[selection:
TLS_RSA_WITH_AES_128_CBC_SHA as defined in RFC 3268
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_ECDHE_RSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA as defined in RFC 4492
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA as defined in RFC 4492
TLS_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
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256 as defined in RFC 5289
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384 as defined in RFC 5289
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256 as defined in RFC 5289
TLS_ECDHE_RSA_WITH_AES_256_GCM_SHA384 as defined in RFC 5289].

Application Note 119
The ciphersuites to be tested in the evaluated configuration are limited by this requirement. The ST author should select the ciphersuites that are supported. It is necessary to limit the ciphersuites that can be used in an evaluated configuration administratively on the server in the test environment. Note that RFC 5246 makes TLS_RSA_WITH_AES_128_CBC_SHA a mandatory ciphersuite, but it is treated as optional for the purposes of conformance with this cPP (i.e. the selection of 'TLS 1.2 (RFC 5246)' will be accepted as conformant with this SFR even if TLS_RSA_WITH_AES_128_CBC_SHA is not one of the ciphersuites listed in the ST).

FCS_TLSS_EXT.1.2 The TSF shall deny connections from clients requesting SSL 1.0, SSL 2.0, SSL 3.0, TLS 1.0, and *[selection: TLS 1.1, TLS 1.2, none].*

Application Note 120
Any TLS versions not selected in FCS_TLSS_EXT.1.1 should be selected here.

FCS_TLSS_EXT.1.3 The TSF shall generate key establishment parameters using RSA with key size 2048 bits and *[selection: 3072 bits, 4096 bits, no other size]* and *[selection: over NIST curves [selection: secp256r1, secp384r1, secp521r1] and no other curves; Diffie-Hellman parameters of size 2048 bits and [selection: 3072 bits, no other size]; no other].*

Application Note 121
The assignments will be filled in based on the assignments performed in FCS_TLSS_EXT.1.1.

8.7 FIA_PMG_EXT.1 Password Management

FIA_PMG_EXT.1 Password management requires the TSF to support passwords with varying composition requirements, minimum lengths, maximum lifetime, and similarity constraints.

Management: FIA_PMG_EXT.1
No management functions.

Audit: FIA_PMG_EXT.1
No specific audit requirements.

Hierarchical to: No other components.
Dependencies: No other components.

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

a) Passwords shall be able to be composed of any combination of upper and lower case letters, numbers, and the following special characters: *[selection: "!", "@", "#", "$", "%", "^", "&", "*", "(", ")", [assignment: other characters]]*;

b) Minimum password length shall be settable by the Security Administrator, and support passwords of 15 characters or greater.

8.8 FIA_UIA_EXT.1 User Identification and Authentication

FIA_UIA_EXT.1 User Identification and Authentication requires administrators (including remote administrators) to be identified and authenticated by the TOE, providing assurance for that end of the communication path. It also ensures that every user is identified and authenticated before the TOE performs any mediated functions.

Management: FIA_UIA_EXT.1
The following actions could be considered for the management functions in FMT:

a) Ability to configure the list of TOE services available before an entity is identified and authenticated

Audit: FIA_UIA_EXT.N
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) All use of the identification and authentication mechanism

b) Provided user identity, origin of the attempt (e.g. IP address)

Hierarchical to: No other components.
Dependencies: FTA_TAB.1 Default TOE Access Banners

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

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

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

8.9 FIA_UAU_EXT.2 Password-based Authentication Mechanism

FIA_UAU_EXT.2 The password-based authentication mechanism provides administrative users a locally based authentication mechanism.

Management: FIA_UAU_EXT.2
The following actions could be considered for the management functions in FMT:

a) None

Audit: FIA_UAU_EXT.2
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: All use of the authentication mechanism

Hierarchical to: No other components.
Dependencies: None

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

8.10 Authentication using X.509 certificates (Extended – FIA_X509_EXT)

FIA_X509_EXT.1 X509 Certificate Validation requires the TSF to check and validate certificates in accordance with the RFCs and rules specified in the component.
FIA_X509_EXT.2 X509 Certificate Authentication, requires the TSF to use certificates to authenticate peers in protocols that support certificates, as well as for integrity verification and potentially other functions that require certificates.
FIA_X509_EXT.3 X509 Certificate Requests, requires the TSF to be able to generate Certificate Request Messages and validate responses.

Management: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3
The following actions could be considered for the management functions in FMT:

a) Remove imported X.509v3 certificates
b) Approve import and removal of X.509v3 certificates
c) Initiate certificate requests

Audit: FIA_X509_EXT.1, FIA_X509_EXT.2, FIA_X509_EXT.3
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Minimal: No specific audit requirements are specified.

FIA_X509_EXT.1 X.509 Certificate Validation

Hierarchical to: No other components
Dependencies: No other components

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 *[selection: the Online Certificate Status Protocol (OCSP) as specified in RFC 2560, a Certificate Revocation List (CRL) as specified in RFC 5280 Section 6.3, Certificate Revocation List (CRL) as specified in RFC 5759 Section 5]*
- The TSF shall validate the extendedKeyUsage field according to the following rules: *[assignment: rules that govern contents of the extendedKeyUsage field that need to be verified].*

Application Note 128
FIA_X509_EXT.1.1 lists the rules for validating certificates. The ST author selects whether revocation status is verified using OCSP or CRLs. The ST author fills in the assignment with rules that may apply to other requirements in the ST. For instance, if a protocol such as TLS that uses certificates is specified in the ST, then certain values for the extendedKeyUsage field (e.g., "Server Authentication Purpose") could be specified.

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.

Application Note 129
This requirement applies to certificates that are used and processed by the TSF and restricts the certificates that may be added as trusted CA certificates.

FIA_X509_EXT.2 X.509 Certificate Authentication

Hierarchical to: No other components
Dependencies: No other components

FIA_X509_EXT.2.1 The TSF shall use X.509v3 certificates as defined by RFC 5280 to support authentication for *[selection: IPsec, TLS, HTTPS, SSH, [assignment: other protocols], no protocols]*, and *[selection: code signing for system software updates, code signing for integrity verification, [assignment: other uses], no additional uses]*.

Application Note 130
If the TOE specifies the implementation of communications protocols that perform peer authentication using certificates, the ST author either selects or assigns the protocols that are specified; otherwise, they select "no protocols". The TOE may also use certificates for other purposes; the second selection and assignment are used to specify these cases.

FIA_X509_EXT.2.2 When the TSF cannot establish a connection to determine the validity of a certificate, the TSF shall *[selection: allow the administrator to choose whether to accept the certificate in these cases, accept the certificate, not accept the certificate].*

Application Note 131
Often a connection must be established to check the revocation status of a certificate - either to download a CRL or to perform a lookup using OCSP. The selection is used to describe the behavior in the event that such a connection cannot be established (for example, due to a network error). If the TOE has determined the certificate valid according to all other rules in FIA_X509_EXT.1, the behavior indicated in the selection determines the validity.

FIA_X509_EXT.3 X.509 Certificate Requests

Hierarchical to: No other components
Dependencies: No other components

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 *[selection: device-specific information, Common Name, Organization, Organizational Unit, Country, [assignment: other information]]*.

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

8.11 FPT_SKP_EXT.1 Protection of TSF Data (for reading of all symmetric keys)

FPT_SKP_EXT.1 Protection of TSF Data (for reading all symmetric keys), requires preventing symmetric keys from being read by any user or subject. It is the only component of this family.

Management: FPT_SKP_EXT.1
The following actions could be considered for the management functions in FMT:
a) There are no management activities foreseen.

Audit: FPT_SKP_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) There are no auditable events foreseen.

Hierarchical to: No other components.
Dependencies: No other components.

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

Application Note 132
The intent of this requirement is for the device to protect keys, key material, and authentication credentials from unauthorized disclosure. This data should only be accessed for the purposes of their assigned security functionality, and there is no need for them to be displayed/accessed at any other time. This requirement does not prevent the device from providing indication that these exist, are in use, or are still valid. It does, however, restrict the reading of the values outright.

FPT_APW_EXT.1 Protection of administrator passwords requires that the TSF prevent plaintext credential data from being read by any user or subject.

Management: FPT_APW_EXT.1
The following actions could be considered for the management functions in FMT:
a) No management functions.

Audit: FPT_APW_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:
a) No audit necessary

Hierarchical to: No other components

Dependencies: No other components.

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.

8.12 FPT_TST_EXT.1 TSF testing

FPT_TST_EXT.1 TSF Self Test requires a suite of self tests to be run during initial start-up in order to demonstrate correct operation of the TSF.

Management: FPT_TST_EXT.1
The following actions could be considered for the management functions in FMT:

a) No management functions.

Audit: FPT_TST_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Indication that TSF self test was completed

FPT_TST_EXT.1.1 The TSF shall run a suite of the following self-tests *[selection: during initial start-up (on power on), periodically during normal operation, at the request of the authorised user, at the conditions [assignment: conditions under which self-tests should occur]]* to demonstrate the correct operation of the TSF: *[assignment: list of self-tests run by the TSF].*

Application Note 133
It is expected that self-tests are carried out during initial start-up (on power on). Other options should only be used if the developer can justify why they are not carried out during initial start-up. It is expected that at least self-tests for verification of the integrity of the firmware and software as well as for the correct operation of cryptographic functions necessary to fulfil the SFRs will be performed. If not all self-test are performed during startup multiple iterations of this SFR are used with the appropriate options selected. In future versions of this cPP the suite of self-tests will be required to contain at least mechanisms for measured boot including self-tests of the components which perform the measurement.

Application Note 134
If certificates are used by the self-test mechanism (e.g. for verification of signatures for integrity verification), certificates are validated in accordance with FIA_X509_EXT.1 and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TST_EXT.2 must be included in the ST.

8.13 FPT_TUD_EXT.1 Trusted update

FPT_TUD_EXT.1 Trusted Update requires management tools be provided to update the TOE firmware and software, including the ability to verify the updates prior to installation.

Management: FPT_TUD_EXT.1
The following actions could be considered for the management functions in FMT:

a) Ability to update the TOE and to verify the updates

b) Ability to update the TOE and to verify the updates using the digital signature capability (FCS_COP.1(2)) and *[selection: no other functions, [assignment: other cryptographic functions (or other functions) used to support the update capability]].*

c) Ability to update the TOE, and to verify the updates using *[selection: digital signature, published hash, no other mechanism]* capability prior to installing those updates

Audit: FPT_TUD_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Initiation of the update process.
b) Any failure to verify the integrity of the update

Hierarchical to: No other components
Dependencies: FCS_COP.1(1) Cryptographic operation (for cryptographic signature), or FCS_COP.1(3) Cryptographic operation (for cryptographic hashing)

FPT_TUD_EXT.1.1 The TSF shall provide Security Administrators the ability to query the currently executing version of the TOE firmware/software and *[selection: the most recently installed version of the TOE firmware/software; no other TOE firmware/software version].*

Application Note 136
The version currently running (being executed) may not be the version most recently installed. For instance, maybe the update was installed but the system requires a reboot before this update will run. Therefore, it needs to be clear that the query should indicate both the most recently executed version as well as the most recently installed update.

FPT_TUD_EXT.1.2 The TSF shall provide *[assignment: authorised users]* the ability to manually initiate updates to TOE firmware/software and *[selection: support automatic checking for updates, support automatic updates, no other update mechanism].*

Application Note 137
The selection in FPT_TUD_EXT.1.2 distinguishes the support of automatic checking for updates and support of automatic updates. The first option refers to a TOE that checks whether a new update is available, communicates this to the administrator (e.g. through a message during an administrator session, through log files) but requires some action by the administrator to actually perform the update. The second option refers to a TOE that checks for updates and automatically installs them upon availability.

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

Application Note 138
The digital signature mechanism referenced in the selection of FPT_TUD_EXT.1.3 is one of the algorithms specified in FCS_COP.1(2).The published hash referenced in FPT_TUD_EXT.1.3 is generated by one of

the functions specified in FCS_COP.1(3). The ST author should choose the mechanism implemented by the TOE; it is acceptable to implement both mechanisms.

Application Note 139
Future versions of this cPP will mandate the use of a digital signature mechanism for trusted updates.

Application Note 140
If certificates are used by the update verification mechanism, certificates are validated in accordance with FIA_X509_EXT.1 and should be selected in FIA_X509_EXT.2.1. Additionally, FPT_TUD_EXT.2 must be included in the ST.

Application Note 141
"Update" in the context of this SFR refers to the process of replacing a non-volatile, system resident software component with another. The former is referred to as the NV image, and the latter is the update image. While the update image is typically newer than the NV image, this is not a requirement. There are legitimate cases where the system owner may want to rollback a component to an older version (e.g. when the component manufacturer releases a faulty update, or when the system relies on an undocumented feature no longer present in the update). Likewise, the owner may want to update with the same version as the NV image to recover from faulty storage. All discrete software components (e.g. applications, drivers, kernel, firmware) of the TSF, should be digitally signed by the corresponding manufacturer and subsequently verified by the mechanism performing the update. Since it is recognized that components may be signed by different manufacturers, it is essential that the update process verify that both the update and NV images were produced by the same manufacturer (e.g. by comparing public keys) or signed by legitimate signing keys (e.g. successful verification of certificates when using X.509 certificates).

8.14 FTA_SSL_EXT.1 TSF-initiated Session Locking

FTA_SSL_EXT.1 TSF-initiated session locking, requires system initiated locking of an interactive session after a specified period of inactivity. It is the only component of this family.

Management: FTA_SSL_EXT.1
The following actions could be considered for the management functions in FMT:

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

Audit: FTA_SSL_EXT.1
The following actions should be auditable if FAU_GEN Security audit data generation is included in the PP/ST:

a) Any attempts at unlocking an interactive session.

Hierarchical to: No other components
Dependencies: FIA_UAU.1 Timing of authentication

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

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

after a Security Administrator-specified time period of inactivity.